

Hospedagem VPS

Debian e Ubuntu



Ribamar FS

2024

<https://ribamar.net.br>

Sumário

Agradecimentos.....	3
Público Alvo.....	4
Sobre o Autor.....	5
Introdução.....	6
Debian e Ubuntu.....	6
1 - Criação do Servidor.....	7
2 - Configuração do DNS.....	9
Adicionar um domínio.....	9
Adicionar registros A.....	9
Sobre Propagação.....	10
Como saber se o domínio propagou?.....	11
3 - Ajustes iniciais do servidor.....	12
Atualizar Servidor.....	12
Ajustar o Timezone.....	12
Criação de uma partição de swap.....	13
4 - Configuração do SSH.....	15
Criar usuário comum.....	15
Adicionar usuário comum ao sudo.....	15
Configurar SSH.....	15
5 - Configuração do Firewall.....	16
Caso precise remover alguma porta use.....	17
Tornando o acesso ao servidor ainda mais seguro.....	17
6 - Instalação e configurações do EMP.....	18
Instalar o Nginx.....	18
Instalar MariaDb.....	18
Suporte ao PHP no Nginx.....	19
7 - Criação dos registros no DNS para cada subdomínio.....	20
8 - Criação dos subdomínios no Nginx.....	21
Testar a sintaxe dos blocos criados.....	23
Reiniciar o Nginx.....	23
9 - Permissões da pasta /var/www.....	24
Podemos tornar o acesso a esta pasta ainda mais restrito.....	24
10 – Configuração do SSL com Certbot.....	26
11 - Criação de usuários e bancos para os aplicativos.....	27
12 – Aplicativos.....	28
13 - Backup.....	30
Backup diário e automático.....	30
Importância do Backup.....	31
14 – Scripts.....	32
15 – Ferramentas.....	34
Grupo de discussão para tirar dúvidas.....	34

Agradecimentos

Sinceramente grato a todos que produzem conhecimento de forma livre, pois mesmo nunca tendo pago um curso nem comprado um bom livro, ainda assim estou hoje elaborando este pequeno manual orientando colegas a criarem e configurarem seus servidores.

Gratidão a todos e que continuemos mantendo a internet com conhecimento.

Público Alvo

Este pequeno manual destina-se a quem está usando uma hospedagem compartilhada e deseja ter mais liberdade e mais controle sobre seu servidor. Com um servidor tipo VPS temos praticamente controle total sobre o servidor. Se quisermos instalara Apache ou Nginx, instalar MySQL ou PostgreSQL, se quisermos instalar qualquer que seja a extensão e versão do PHP. É você quem decide. Mas é importante saber que todo o gerenciamento é por sua conta, instalar e configurar um firewall, configurar o SSH e numa porta desejada. O suporte oferecido pelo serviço que contratamos praticamente deve ser esquecido, você está por sua conta.

Isso que falei acima mostra que se você quer ter conforto deve ficar com as hospedagens compartilhadas, mas se tem disposição para aprender e quer pagar o preço para ter um grande controle sobre seu servidor, então estamos a qui para ajudá-lo. Se decidir usar VPS lembre-se que precisa ser um autodidata e estudar muito. Não desista nas primeiras dificuldades e se tornará um profissional melhor e mais valorizado no mercado.

Sobre o Autor

Sou o ribafs/Ribamar FS, um apaixonado pela programação web com PHP, por Linux e pela administração de servidores linux.

Participo ativamente em diversos grupos de discussão no Facebook, tanto que em geral sou classificado de supercolaborador.

Estudo e trabalho com TI há uns 30 anos.

Um currículo não atualizado

<https://ribafs.github.io/sobre/curriculo/>

Meu site atual, criado num VPS do DigitalOcean. Meu pequeno laboratório criado com Joomla:

<https://ribamar.net.br/>

Outros livros meus (todos free)

<https://ribafs.github.io/sobre/livros/>

Introdução

Este treinamento está sendo publicado de uma forma diferente. Inteiramente gratuito. Apenas com sugestão de doação. Mas, sinceramente, gostaria de receber doações somente de quem sentir vontade de fazê-lo, daqueles que perceberem que está sendo útil e está podendo fazê-lo.

Um livro bem prático, com receitas de bolo e pouca teoria.

Debian e Ubuntu

O Ubuntu é uma distribuição oriunda do Debian, que revolucionou o desktop linux. Atualmente é uma das mais populares distribuições para desktop e também para servidor.

No desktop ele ficou bem diferente do Debian, mas quando instalamos o servidor em um serviço VPS a diferença entre Debian e Linux é bem pequena, especialmente nos procedimentos de configuração. Os procedimentos básicos para uma configuração do Debian podem sem mudança ser aplicados para uma configuração do Ubuntu sem alteração. Apenas que o Ubuntu já traz pré-instalados alguns pacotes e facilita um pouco algumas tarefas.

O foco do Ubuntu é nos usuários menos experientes, portanto facilita tarefas e já vem com alguns pacotes instalados que o Debian não traz. Ao final de uma instalação básica percebo que o Debian ocupa um espaço menos em disco e consome menos memória RAM e isso me faz preferir o Debian, especialmente com servidor usando poucos recursos. Por isso aqui os procedimentos são para o Debian e apenas quando lembrar de uma diferença que vale o comentário eu comentarei, ok?

1 - Criação do Servidor

Lembrando que apenas os passos da criação do servidor e da configuração do DNS são específicos do serviço de VPS. Neste caso usei o DigitalOcean, mas podemos usar qualquer servidor, pois a parte mais substancial e mais importante é a configuração dos serviços e instalação do EMP.

Criação e dimensionamento do servidor

Este item vai depender das necessidades do servidor. Alguns podem ser bem simples e outros com muito recurso.

Selecionei este servidor para criar meus sites e aplicativos

1 GB Memory

1 AMD vCPU CPU-Optimized

25 GB Disk

NYC1

Debian 12 x64

7/mo

Hostname – RibaFS

IP - 64.23.165.38

Após a criação do servidor é muito importante anotar o seu IP, especialmente antes que o domínio propague.

SSH

Minha recomendação é que não use uma chave do SSH na criação do servidor, mas sim use senha, para que assim possa acessar o servidor de outros computadores e mesmo se vier a formatar o seu próprio continue acessando dele. Caso use uma chave precisará tomar outros cuidados, como backup da chave e/ou outros.

Criação do Servidor no DigitalOcean

De forma resumida

Choose Region/Datacenter

Escolhi San Francisco

Choose an image

Debian – 12 x64

Choose size

Basic

Choose Authentication Method

Password

Entre com a senha do root (veja abaixo as exigências)

Hostname

debians

Tags

debian

Project

RibaFS

Create Droplet

Aguardar a criação para capturar o IP

Se precisar de ajuda para se cadastrar no DigitalOcean e criar seu primeiro servidor veja este artigo:

<https://ribamar.net.br/index.php/servidores/vps/servidor-no-digitalocean>

Debian 12 traz nativamente pacotes:

- PHP 8.2
- Nginx 1.2
- MariaDb 10

Isso nos permite instalar os aplicativos mais recentes.exemplo:

- Joomla 5.0.2
- Laravel 10 e o próximo 11

Entre outros

2 - Configuração do DNS

Setar DNS

No DigitalOcean para ter acesso ao painel de configuração do DNS

Manage

Networking

Domains

Observação: Lembrando que em cada serviço de hospedagem a configuração do DNS é diferente.

Adicionar um domínio

Entre com o domínio na caixa Enter domain e clique em Add domain. Observe que quando adicionamos um domínio ele nos mostra os Nameservers (NS) da hospedagem, no caso:

NS	ribamar.net.br	directs to ns1.digitalocean.com.
NS	ribamar.net.br	directs to ns2.digitalocean.com.
NS	ribamar.net.br	directs to ns3.digitalocean.com.

Precisamos anotar estes NS: ns1.digitalocean.com, ns2.digitalocean.com e ns3.digitalocean.com e levar até a administração do nosso domínio (meu caro o registro.br e minha recomendação) para apontar nosso domínio para que saiba onde encontrar nosso site.

Dúvidas? Pergunte em nosso grupo do FB: <https://www.facebook.com/groups/ribafsvps>

Adicionar registros A

Selecionar o domínio para o qual deseja adicionar registros.

Observação importante: aqui iremos adicionar apenas os registros básicos de um domínio, não trabalharemos com registros para servidores de e-mail nem outros. Criaremos para cada subdomínio dois registros, como veremos

A ribamar.net.br 64.23.165.38

A www.ribamar.net.br 64.23.165.38

Create new record

A AAAA CNAME MX TXT NS SRV CAA

Use @ to create the record at the root of the domain or enter a hostname to create it elsewhere. A records are for IPv4 addresses only and tell a request where your domain should direct to.

HOSTNAME	WILL DIRECT TO	TTL (SECONDS)	
<input type="text" value="Enter @ or hostname"/>	<input type="text" value="Select resource or enter custom IP"/>	<input type="text" value="Enter TTL 3600"/>	<input type="button" value="Create Record"/>

Veja que o registro A já vem selecionado por padrão.

Vamos dizer ao DNS que nosso domínio principal é o `ribamar.net.br`. Para isso na caixa HOSTNAME digitamos @ e na caixa WILL DIRECT TO clicamos e selecionamos o IP do servidor que criamos, assim:

Então clicar em Create Record

HOSTNAME	WILL DIRECT TO	TTL (SECONDS)	
<input type="text" value="Enter @ or hostname"/> @	<input type="text" value="portal-debian SYD1 / 170.64.204.218"/>	<input type="text" value="Enter TTL 3600"/>	<input type="button" value="Create Record"/>

ribamar.net.br

Adicionar um registro A para o subdomínio www do nosso domínio.

Como muita gente está acostumada a digitar o `www` no início do domínio e não queremos que estas pessoas se deparem com um erro de 404, então vamos adicionar o subdomínio `www`:

HOSTNAME	WILL DIRECT TO	TTL (SECONDS)	
<input type="text" value="Enter @ or hostname"/> www	<input type="text" value="portal-debian SYD1 / 170.64.204.218"/>	<input type="text" value="Enter TTL 3600"/>	<input type="button" value="Create Record"/>

www.ribamar.net.br

Veja que antes mesmo de clicar no botão ele já mostra o resultado: www.ribamar.net.br.
Clicar em Create Register.

Claro que você deve trocar o domínio e o IP para os seus.

Sobre Propagação

O DNS é uma ferramenta da internet que faz a ligação de um IP com um domínio. Assim as pessoas não precisam memorizar números, mas apenas os nomes dos domínios.

Acontece que após apontar os NS da hospedagem na administração do domínio acontece algo chamado de propagação. Precisamos esperar de alguns minutos até 72 horas para que o nome do domínio seja ligado pelo DNS

Como saber se o domínio propagou?

No linux eu gosto de usar um comando no terminal para saber se o domínio já propagou:

```
host ribamar.net.br
```

Caso já tenha propagado ele me mostra o domínio e o IP associado a ele.

```
ribamar.net.br has address 64.23.165.38
```

Também existem alguns serviços online que mostram ainda mais detalhes.
O que mais uso é este:

<https://www.whatsmydns.net/>

Você digita o domínio e clica em Search. Veja que o registro tipo A já está selecionado.

Obs: Para que configuremos o SSL com o Certbot, há necessidade de o domínio já ter propagado.

3 - Ajustes iniciais do servidor

Tendo em mãos o IP fornecido pelo serviço após a criação do servidor, abrimos um terminal e executamos:

```
ssh root@64.23.165.38
```

Na primeira conexão ele mostrará:

```
The authenticity of host '64.23.165.38 (64.23.165.38)' can't be established.  
ED25519 key fingerprint is SHA256:UWy6RME4tCB8hv2qWYVEhajZ8seianetnSxahDdcO24.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Digite yes e Enter

Forneça a senha usada na criação do servidor.

Observe que no primeiro acesso estamos acessando com o usuário root, que não precisa de sudo para executar tarefas privilegiadas. Mais a frente estaremos criando um usuário comum e dando direito a ele de usar o sudo e daí em diante administraremos nosso servidor com este user e não mais com o root, visto que o root é muito visado e tem muitos poderes.

Atualizar Servidor

Vamos atualizar os repositórios e, caso existam, também os pacotes:

```
apt update;apt upgrade -y
```

Aguarde...

O -y ao final faz com que ele atualize sem nos perguntara. Aguarde...

Caso apareça o aviso

```
What do you want to do about modified configuration file sshd_config?
```

Recomendo que selecione a opção
install the package maintainer's version
Que é a primeira

Para que substitua a versão existente com a nova versão.

Ajustar o Timezone

```
timedatectl set-timezone "America/Fortaleza"
```

Altere para sua região.

Verificar o resultado

```
timedatectl
```

Referência

<https://tecmint.com/2016/05/change-timezone-on-debian/>

Verificar o espaço em disco e a memória RAM

```
df -h
```

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1        9.7G  1.7G  7.6G  19% /
```

```
free -m
```

```
free -m
              total        used         free       shared  buff/cache   available
Mem:           457           105          159            3         208         352
Swap:            0             0             0
```

Veja que não tem swap e tem bem pouca RAM, somente 512 MB. Então criamos uma partição de swap com 1GB.

Criação de uma partição de swap

Especialmente importante para servidores com 1GB de RAM ou menos.

```
fallocate -l 1G /swapfile
chmod 600 /swapfile
mkswap /swapfile
swapon /swapfile
```

```
nano /etc/fstab
Adicione ao final
```

```
/swapfile swap swap defaults 0 0
```

Execute novamente o comando

```
free -m
              total        used         free       shared  buff/cache   available
Mem:           457           105          155            3         212         352
Swap:          1023             0          1023
```

Quando o servidor começar a usar swap é sinal de que precisa melhorar os recursos de RAM e processamento.

Veja a partição de swap criada. Agora, quando a memória RAM estiver sendo usada totalmente, o sistema poderá usar a partição de swap e não travará. Idealmente, se seu sistema estiver usando swap é recomendado aumentar a memória RAM, pois o uso da swap deixa o servidor mais lento,

Reiniciar

reboot

Raramente precisamos reiniciar um linux, mas neste caso, geralmente temos a instalação de novo kernel, então reiniciamos para que o novo kernel seja carregado.

Aguarde em torno de um minuto para voltar a se conectar com root.

4 - Configuração do SSH

O acesso ao servidor via SSH já é normalmente seguro, mas iremos fazer alguns poucos ajustes para torná-lo ainda mais seguro e idealmente ficar acessando o servidor somente com um usuário comum e não mais com o root. Aqui estou sugerindo trocar a porta e a recomendação é que troquemos por uma porta alta, maior que a 1024. Mude para a que achar melhor.

Acessar novamente usando root

```
ssh root@64.23.165.38
```

clear

Criar usuário comum

```
adduser ribafs
```

Acima somos obrigados a digitar a senha, repetir, o nome e para o restante podemos apenas teclar Enter.

Adicionar usuário comum ao sudo

```
adduser ribafs sudo
```

Configurar SSH

```
nano /etc/ssh/sshd_config
```

```
Port 60022
```

```
LoginGraceTime 30
```

```
PermitRootLogin no
```

```
MaxAuthTries 3
```

```
X11Forwarding no
```

```
AllowUsers ribafs
```

Para ficar mais prático pode adicionar estas linhas ao final do arquivo. Assim elas prevalecerão sobre as anteriores.

Reiniciando o SSH para que as alterações tenham efeito

```
service ssh restart
```

```
exit
```

Conectar com ribafs

Referências

<https://kifarunix.com/allow-deny-specific-users-to-login-via-ssh-on-ubuntu-18-04/>

5 - Configuração do Firewall

A partir de agora já devemos usar o usuário que criamos e executar os comandos com sudo. Veja também que mudamos a porta 22, que é a padrão, para a 60022, então o acesso agora é com:

```
ssh -p 60022 ribafs@64.23.165.38
```

```
clear
```

Agora, sempre que formos executar um comando que precise de privilégios elevados precisamos iniciar com sudo.

No caso do Ubuntu ele já vem com o ufw instalado, mas no Debian precisamos instalar:

```
sudo apt install ufw
```

Ele pedirá a senha do user ribafs.

Vamos liberar somente as portas SSH, HTTP e HTTPS

```
sudo ufw allow 60022
```

```
sudo ufw allow 80
```

```
sudo ufw allow 443
```

Adicionamos as portas e agora vamos habilitar o firewall:

```
sudo ufw enable
```

Confirme digitando y e enter

Vejamos como está:

```
sudo ufw status
```

```
Status: active
```

To	Action	From
--	-----	----
60022	ALLOW	Anywhere
80	ALLOW	Anywhere
443	ALLOW	Anywhere
60022 (v6)	ALLOW	Anywhere (v6)
80 (v6)	ALLOW	Anywhere (v6)
443 (v6)	ALLOW	Anywhere (v6)

Caso precise remover alguma porta use

sudo ufw status numbered

sudo ufw delete 1

Tornando o acesso ao servidor ainda mais seguro

Mudar a primeira linha para

ufw allow from 177.14.224.188 to any port 60022

Assim o usuário somente conseguirá acessar o servidor via SSH se seu IP externo for o 177.14.224.188.

Como descobrir seu IP externo

<https://meuip.com.br/>

6 - Instalação e configurações do EMP

Logar com ribafs

```
ssh -p 60022 ribafs@64.23.165.38
```

Atualizar repositórios e pacotes

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

Instalar o Nginx

```
sudo apt install nginx -y
```

Acesse no navegador

<http://64.23.165.38>

Instalar MariaDb

```
sudo apt install mariadb-server -y
```

Melhorar a segurança do MariaDb

```
sudo mysql_secure_installation
```

Enter current password for root (enter for none):

Enter

Switch to unix_socket authentication [Y/n]

Y ou apenas Enter (se Y está em maiúscula basta enter para Y)

Change the root password? [Y/n]

Y

Entre com uma senha e repita

Remove anonymous users? [Y/n]

Y

Disallow root login remotely? [Y/n]

Y

Remove test database and access to it? [Y/n]

Y

Reload privilege tables now? [Y/n]

Y

Testando

```
mysql -uroot -p
```

Suporte ao PHP no Nginx

Para adicionar suporte ao PHP no Nginx, instalar e usar o PHP-FPM para executar arquivos PHP

```
sudo apt install aptitude git composer mc curl phpunit ssh imagemagick zip unzip -y
```

```
sudo apt install php-fpm php-mysql php-gd php-cli php-curl php-mbstring php-zip php-opcache  
php-xml php-bcmath php-pear php-imagick php-tidy php-xmlrpc php-intl php-xdebug php-apcu  
php-redis -y
```

```
sudo mkdir -p /etc/apt/keyrings  
curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o  
/etc/apt/keyrings/nodesource.gpg
```

```
NODE_MAJOR=20  
echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg]  
https://deb.nodesource.com/node_${NODE_MAJOR}.x nodistro main" | sudo tee  
/etc/apt/sources.list.d/nodesource.list  
sudo apt update  
sudo apt install nodejs -y  
sudo apt update && sudo apt install -y yarn
```

Caso tenha dificuldade em copiar estes comandos veja o script para download deste e-book

<https://ribamar.net.br/vps/emp.sh>

Anotar

```
/etc/php/8.2/fpm/php.ini
```

Ver as versões

```
php -v
```

```
node -v
```

7 - Criação dos registros no DNS para cada subdomínio

Iremos criar os subdomínios:

default – já está criado por padrão

joomla.ribamar.net.br

laravel.ribamar.net.br

Com estas informações seguimos ao painel de administração do DNS e criamos dois registros A para cada subdomínio, de forma que fique assim:

default

A ribamar.net.br 64.23.165.38

A [www.ribamar.net](http://www.ribamar.net.br),br 64.23.165.38 (para criar este apenas digitamos www)

joomla

A joomla.ribamar.net.br 64.23.165.38

A www.joomla.ribamar.net.br 64.23.165.38

laravel

A laravel.ribamar.net.br 64.23.165.38

A www.laravel.ribamar.net.br 64.23.165.38

8 - Criação dos subdomínios no Nginx

Iremos agora criar os subdomínios no Nginx para o capítulo anterior:

default - /var/www/html – ribamar.net.br e www.ribamar.net.br (aqui o wordpress)

joomla - /var/www/joomla – joomla.ribamar.net.br e www.joomla.ribamar.net.br

laravel - /var/www/laravel – laravel.ribamar.net.br e www.laravel.ribamar.net.br

```
cd /etc/nginx/sites-available
```

```
sudo nano default
```

```
server {
    listen 80;
    listen [::]:80;
    root /var/www/html;
    index index.html index.php;

    server_name ribamar.net.br www.ribamar.net.br;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.2-fpm.sock;
    }
}
```

```
sudo nano joomla.ribamar.net.br
```

```
server {
    listen 80;
    listen [::]:80;
    root /var/www/joomla;
    index index.html index.php;
    server_name joomla.ribamar.net.br www.joomla.ribamar.net.br;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.2-fpm.sock;
    }
}
```

```
}  
}
```

```
sudo ln -s /etc/nginx/sites-available/joomla.ribamar.net.br /etc/nginx/sites-enabled/joomla.ribamar.net.br
```

Observe que o default não precisa de link simbólico.

```
sudo nano laravel.ribamar.net.br
```

```
server {  
    listen 80;  
    listen [::]:80;  
    root /var/www/laravel/public;  
    index index.html index.php;  
    server_name laravel.ribamar.net.br www.laravel.ribamar.net.br;  
  
    location / {  
        try_files $uri $uri/ /index.php?$args;  
    }  
    location ~ \.php$ {  
        include snippets/fastcgi-php.conf;  
        fastcgi_pass unix:/run/php/php8.2-fpm.sock;  
    }  
}
```

```
sudo ln -s /etc/nginx/sites-available/laravel.ribamar.net.br /etc/nginx/sites-enabled/laravel.ribamar.net.br
```

Caso queira adicionar um segundo domínio neste mesmo servidor, precisa adicionar seus dois registros A ao DNS e adicionar o bloco/subdomínio ao Nginx, como a seguir:

```
sudo nano fatima.adv.br
```

```
server {  
    listen 80;  
    listen [::]:80;  
    root /var/www/fatima;  
    index index.html index.php;  
  
    server_name fatima.adv.br www.fatima.adv.br;  
  
    location / {  
        try_files $uri $uri/ /index.php?$args;  
    }  
    location ~ \.php$ {  
        include snippets/fastcgi-php.conf;  
        fastcgi_pass unix:/run/php/php8.2-fpm.sock;  
    }  
}
```

```
sudo ln -s /etc/nginx/sites-available/fatima.adv.br /etc/nginx/sites-enabled/fatima.adv.br
```

Novamente lembro que se tiver dificuldade para copiar estes script faça o download do arquivo em

<https://ribamar.net.br/vps/subdominios.txt>

Testar a sintaxe dos blocos criados

```
sudo nginx -t
```

Reiniciar o Nginx

```
sudo systemctl restart php8.2-fpm nginx
```

Se tudo ok não mostra nenhuma mensagem na tela.

Antes de acessar espere que ajustemos as permissões e que instalemos o SSL.

9 - Permissões da pasta /var/www

Setar as permissões dos arquivos da pasta /var/www

Ao copiar ou criar arquivos/pastas dentro da pasta (exemplo: /var/www/html) o(s) arquivos herdarão as permissões da pasta.

Evite mover arquivos para o /var/www, pois assim eles herdam as permissões originais e precisará executar o script perms.

Adicionar seu user para o grupo www-data

```
sudo adduser ribafs www-data
```

```
sudo nano /usr/local/bin/perms
```

```
#!/bin/sh
clear;
echo "Aguarde enquanto configuro as permissões do /var/www/$1";
echo "";
chown -R ribafs:www-data /var/www/$1;
find /var/www/$1 -type d -exec chmod 775 {} \;
find /var/www/$1 -type f -exec chmod 664 {} \;
find /var/www/$1 -type d -exec chmod g+s {} \;
echo "";
echo "Concluído!";
```

```
sudo chmod +x /usr/local/bin/perms
```

Usando

```
sudo perms (varre toda a /var/www)
```

```
sudo perms html(varre /var/www/html)
```

Podemos tornar o acesso a esta pasta ainda mais restrito

Para isso criamos um usuário sem shell (ribafsw), que não tem acesso ao servidor via SSH e configuramos a pasta para que seu dono seja este usuário.

Criar o usuário

```
sudo adduser ribafsw --no-create-home --shell /usr/sbin/nologin
```

Adicionamos ele ao grupo www-data, que é o grupo do Nginx

```
sudo adduser ribafsw www-data
```

Criamos o novo script permsw

```
nano /usr/local/bin/permsw
```

```
#!/bin/sh
clear;
echo "Aguarde enquanto configuro as permissões do /var/www/$1";
echo "";
chown -R ribafsw:www-data /var/www/$1;
find /var/www/$1 -type d -exec chmod 755 {} \;
find /var/www/$1 -type f -exec chmod 644 {} \;
find /var/www/$1 -type d -exec chmod g+s {} \;
echo "";
echo "Concluído!";
```

Arquivos copiados para /var/www ou criados nesta pasta ou subs recebem as mesmas permissões existentes

Assim ribafs não tem permissão de mexer em /var/www
Somente terá se eu mudar para 775 e 664, que darão permissão também ao grupo

Com isso os usuários atuais dos sites e aplicativos não poderão alterar os mesmos, somente acessar.
Caso queira que os usuários tenham acesso irrestrito aos sites e aplicativos precisará rodar o script

```
sudo perms site
```

Que torna o ribafs dono dos mesmos.

10 – Configuração do SSL com Certbot

O Certbot é uma ótima opção para adicionar SSL free no Servidor. Ótimo não somente por ser free mas por ser bem prático de instalar.

Acessamos o site
<https://certbot.eff.org/>

Rolamos um pouco a tela e em

Software selecionamos Nginx
System selecionamos Debian Testing

Rolar a tela e seguir os passos.

<https://certbot.eff.org/instructions?ws=nginx&os=debiantesting>

Já tenho aqui os comandos então:

```
sudo apt update
```

```
sudo apt install snapd -y
```

```
sudo snap install --classic certbot
```

```
sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

Configurar o SSL

```
sudo certbot --nginx
```

Entre com seu e-mail

Y

Y

Enter

para todos

Renovação automática

```
sudo certbot renew --dry-run
```

```
sudo nginx -t
```

```
sudo systemctl restart php8.2-fpm nginx
```

11 - Criação de usuários e bancos para os aplicativos

Criaremos 3 bancos e seus 3 respectivos usuários:

wp - wordpress
joomla
laravel

sudo mysql

```
create database joomla_db CHARACTER SET utf8 COLLATE utf8_general_ci;  
CREATE USER 'joomla_us'@'localhost' IDENTIFIED BY 'abcd1234J@';  
GRANT ALL PRIVILEGES ON joomla_db.* TO 'joomla_us'@'localhost' WITH GRANT OPTION;
```

```
create database laravel_db CHARACTER SET utf8 COLLATE utf8_general_ci;  
CREATE USER 'laravel_us'@'localhost' IDENTIFIED BY 'abcd1234L@';  
GRANT ALL PRIVILEGES ON laravel_db.* TO 'laravel_us'@'localhost' WITH GRANT OPTION;
```

```
create database wp_db CHARACTER SET utf8 COLLATE utf8_general_ci;  
CREATE USER 'wp_us'@'localhost' IDENTIFIED BY 'abcd1234W@';  
GRANT ALL PRIVILEGES ON wp_db.* TO 'wp_us'@'localhost' WITH GRANT OPTION;
```

Aqui também temos um arquivo para download

<https://ribamar.net.br/vps/bancos.txt>

12 – Aplicativos

Existe uma grande quantidade de bons aplicativos que podemos instalar com suporte ao PHP.

Os que mais tenho usado são o CMS Joomla e o framework laravel.

Joomla

Instalação

Baixar a última versão do site oficial

<https://www.joomla.org/>

Ou outra versão, como a 4.4.2 do repositório no Github:

<https://github.com/joomla/joomla-cms/releases>

Acesse o servidor e na pasta /var/www execute:

```
wget linkdozip
```

```
unzip linkdozip -d /var/www/joomla
```

Após descompactar edecute

```
sudo perms joomla
```

Wordpress

Acesse

<https://br.wordpress.org/download/>

Role um pouco a tela e copie o link para download para a memória

```
cd /var/www
```

```
wget https://br.wordpress.org/latest-pt\_BR.zip
```

```
unzip latest-pt_BR.zip -d /var/www/html
```

Então acesse via web e instale. Lembre de antes criar banco e respectivo user para ele.

Laravel

Aqui temos uma particularidade, ele deve ser descompactado não em /var/www/laravel, mas na pasta public /var/www/laravel/public.

Este é melhor instalado via composer, pela linha de comando.

Citando outros aplicativos com PHP

CMS Wordpress

Framework CodeIgniter

Framework Symfony

CMS Drupal

CMS Moodle (EAD)

Loja virtual OpenCart

Loja virtual com Laravel - <https://aimeos.org/help/>

13 - Backup

Uma das providências mais importantes em termos de segurança na web, backup dos arquivos e do banco de dados. A frequência deve ser diretamente proporcional à importância do site ou aplicativo.

Um exemplo de script que uso para efetuar backup de um site:

```
mysqldump -uroot -pzmxn1029M@ portal_db > /home/ribafs/lasts/portal_$(date +%Y_%m_%d").sql  
  
cd /var/www/  
zip -rq /home/ribafs/lasts/portal_$(date +%Y_%m_%d").zip html
```

Caso queira deixar de fora alguma das subpastas do site use:

```
zip -rq /home/ribafs/lasts/portal_$(date +%Y_%m_%d").zip html -x "html/down/*"
```

Veja que eu salvo na pasta /home/ribafs/lasts e depois baixo dela para o meu desktop.

Com este sufixo

```
_$(date +%Y_%m_%d")
```

Temos um backup por dia. Caso queira uma frequência maior, use algo como:

```
_$(date +%Y_%m_%d:%H:%i")
```

Backup diário e automático

Para isso podemos usar o cron

Criar um script para backup em /home/ribafs/backupdia

Dar permissão de execução para ele:

```
chmod +x /home/ribafs/backupdia
```

```
crontab -e
```

Adicionar ao final:

```
0 2 0 0 0 /home/ribafs/backupdia
```

Assim ele fará automaticamente o backup todos os dias as 2 horas da manhã.

O script *backupdia* deve conter tudo que desejamos fazer o backup diariamente. Depois veremos como copiar o conteúdo da pasta lasts para o desktop.

Importância do Backup

Ao meu ver o backup é uma das providências mais importantes em termos de segurança, pois se por algum motivo perder o site, poderá então restaurar um backup recente e confiável. Mesmo que não consiga acesso ao servidor poderá restaurar em outro.

Lembre de manter pelo menos umas 3 cópias dos sites e aplicativos e é muito importante que teste o restore em seu desktop e em outra pasta do servidor.

14 – Scripts

Para tornar mais prática a administração do servidor é útil criar alguns scripts e aliases.

O shell script é um verdadeiro canivete suíço, com uma grande gama de recursos que podem tornar a vida do administrador do servidor bem mais amena.

Aliases

Na pasta do usuário

```
cd /home/ribafs
```

```
nano .bashrc
```

Adicione ao final do arquivo:

```
alias c="clear"  
alias e="exit"  
alias cw="cd /var/www"  
alias nr="sudo systemctl restart php8.2-fpm nginx"  
# Executar o backup manualmente  
alias backup="sh /home/ribafs/backupdiario"  
alias s="sudo su"  
alias rc="nano .bashrc"  
alias rcs="source ~/.bashrc"  
alias phpI="sudo nano /etc/php/8.2/fpm/php.ini;sudo systemctl restart php8.2-fpm nginx"  
clear
```

O último clear, fará com que o servidor, após ser acessado via SSH, limpe a tela.

Aliases no desktop que agem com o servidor

Adicionar ao final do .bashrc do usuário do desktop

```
# Trazer para a pasta atual do desktop todos os arquivos da pasta /home/ribafs/lasts do servidor  
alias copiars="scp -P 60022 ribafs@ribamar.net.br:/home/ribafs/lasts/* ."  
alias rc="nano .bashrc"  
alias rcs="source ~/.bashrc"  
# Acessar o servidor via SSH  
alias serv="ssh -p 60022 ribafs@ribamar.net.br"  
# Mostrar maiores arquivos da pasta atual em ordem decrescente  
alias maiores="du -h | egrep -v "\\.+/" | sort -hr"
```

backupdiario

```
/home/ribafs/backupdiario
```

```
rm -rf /home/ribafs/lasts/*
```

```
mysqldump -uroot -pzmxn1029M@ portal_db > /home/ribafs/lasts/portal_$(date +"%Y_%m_%d").sql
```

```
cd /var/www/
```

```
#zip -rq /home/ribafs/lasts/portal_$(date +"%Y_%m_%d").zip html -x "html/down/*"
```

```
zip -rq /home/ribafs/lasts/portal_$(date +"%Y_%m_%d").zip html
```

```
cp -f /home/ribafs/backupdiario /home/ribafs/lasts
```

Veja que este se preocupa em não lotar o disco do servidor, apagando todos os arquivos da pasta lasts logo no início. Caso esteja em um servidor com muito espaço em disco e cujo backup seja bem importante pode comentar essa linha.

Scripts no desktop

Para ficarem no path eu gosto de criar na pasta /usr/local/bin

scpe – enviar arquivos do desktop para o servidor usando o SCP

```
#De - $1 (servidor)
```

```
#Para - $1 (local)
```

```
scp -P 65522 $1 ribafs@ribamar.net.br:$2"
```

Exemplo:

Enviar o arquivo /home/ribafs/arq.zip para a pasta /var/www do servidor

```
scpe /home/ribafs/arq.zip /var/www
```

scpr – receber arquivos de uma pasta no servidor numa pasta do desktop

```
#De - $1 (servidor)
```

```
#Para - aqui (local)
```

```
scp -P 65522 ribafs@ribamar.net.br:$1 $2
```

Exemplo

Receber o arquivo /var/www/portal.zip na pasta atual

```
scpr /var/www/portal.zip .
```

15 – Ferramentas

Gostaria de apresentar/sugerir apenas um gerenciador de bancos de dados que uso, é muito prático e pouco conhecido, que é o adminer.

Ele suporta
mysql/mariadb
postgresql
sqlite
oracle
mssql
mongodb
elasticsearch

Em apenas um único arquivo php

<https://www.adminer.org/>

Podemos baixar ele na pasta de algum dos subdomínios e de preferência renomear.

Grupo de discussão para tirar dúvidas

<https://www.facebook.com/groups/ribafsvps>

Observação importante

Não sou patrocinado em nada pelo DigitalOcean, apenas me refiro a eles por conta de que meu servidor está com eles. Meu servidor está com eles, por conta de eu os ter selecionado entre os outros que experimentei.

Fortaleza, 17 de janeiro de 2024
Ribamar FS