

Virtual Hosts, virtualizando vários sites em um mesmo servidor



Yuri Matheus

04/09/2017



Formação DevOps

Precisei virtualizar dois sites diretamente de um servidor Apache, atualmente temos a seguinte arquitetura nos servidores:

Hardware

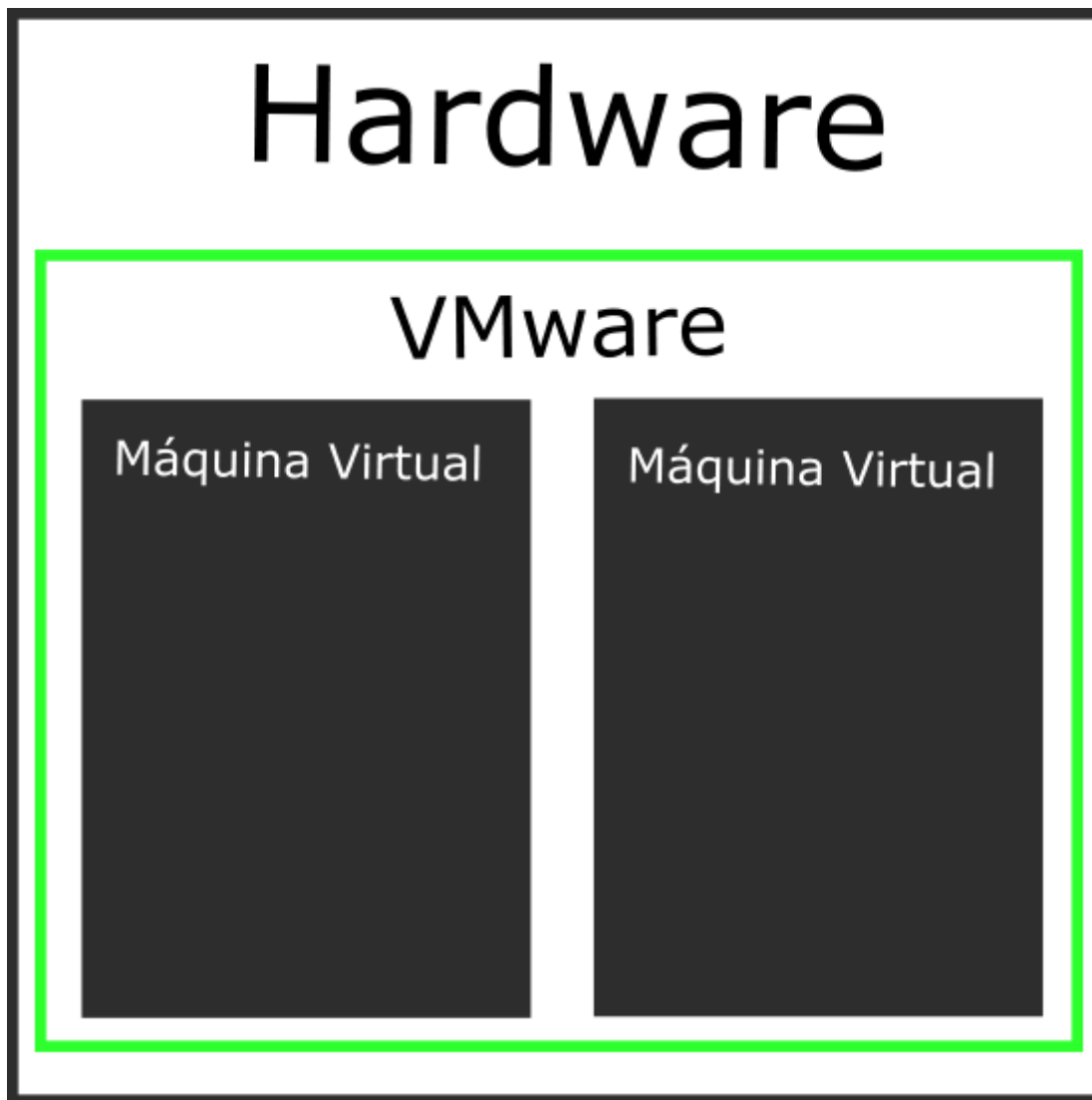
Sistema Operacional

Apache Servidor Web

Note que temos apenas um sistema operacional com um **Apache** instalado que, por padrão, só roda um site de cada vez.

Então como hospedar mais de um site, se existe apenas uma máquina?

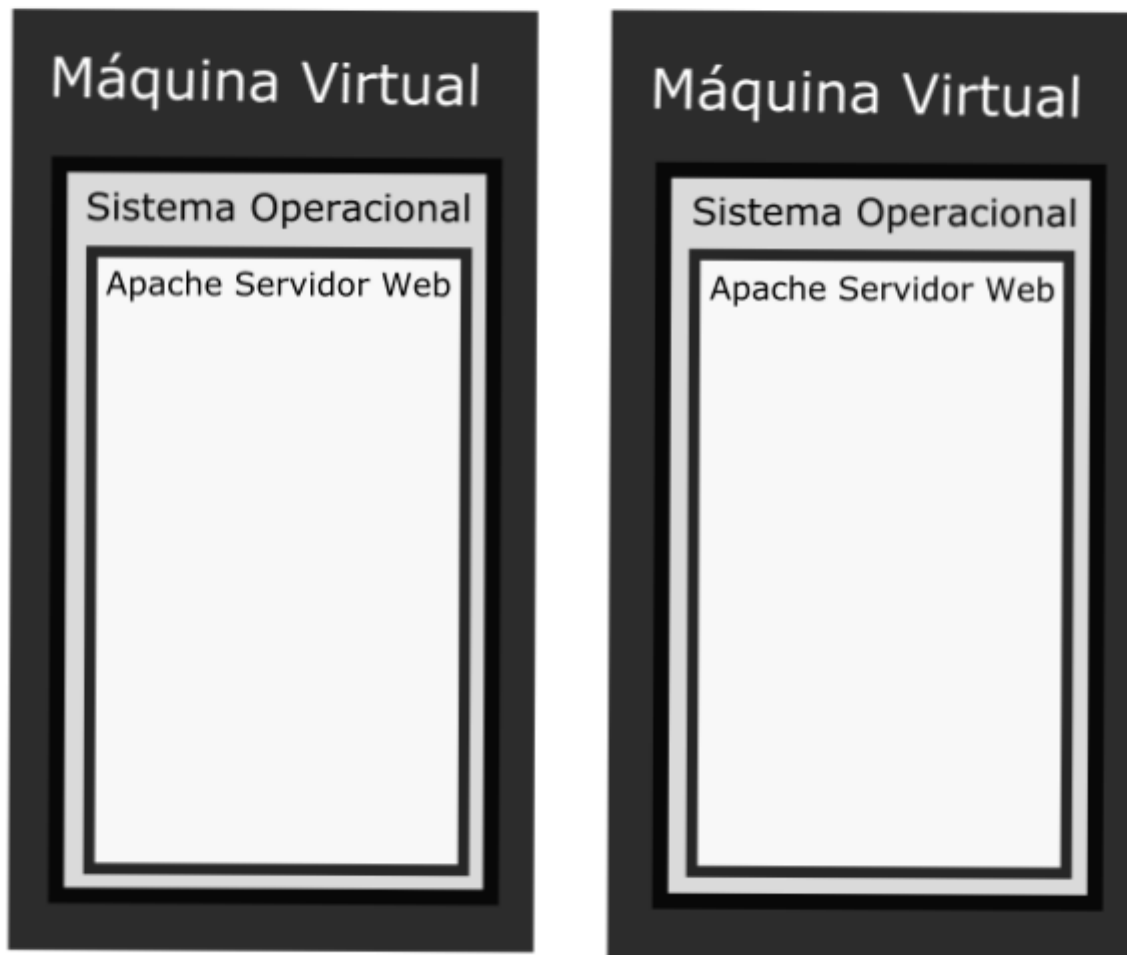
Uma possibilidade é utilizar um virtualizador, como o [VMware](#), que possibilita que outros sistemas operacionais rodem em uma única máquina, e colocar para rodar dois [servidores virtuais](#):



Assim continuaríamos com uma máquina física, porém com dois servidores virtuais. Cada um para um site:

Hardware

VMware



Porém teríamos que configurar o software de virtualização, instalar novamente o sistema operacional do servidor e configurar dois servidores. Além da virtualização consumir mais recursos do hardware.

Em outras palavras, o custo para o consumidor ficaria mais caro, já que precisaria investir um pouco mais de tempo em melhorias no hardware.

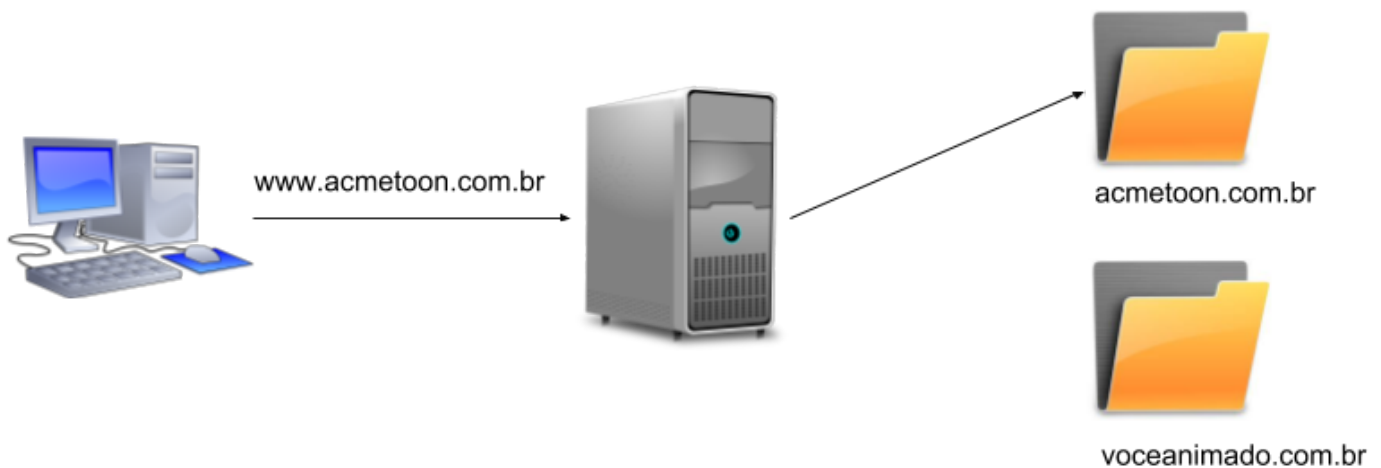
Bem, para evitar isso, existe um recurso que nos permite virtualizar vários sites diretamente do mesmo servidor utilizando o **Apache** (ou outro servidor web, como o

[nginx](#)), por exemplo, ele se chama [Virtual Hosts](#).

Entendendo os Virtual Hosts

Os hosts virtuais funcionam dessa maneira:

- Um computador faz a requisição para acessar um site pelo seu nome de domínio;
- Esse pedido chega ao nosso servidor web que encaminha para o host virtual da requisição, podendo ser uma pasta ou outro servidor:



Como pode ser visto, o servidor recebe a requisição e encaminha para o diretório que está configurado.

Sabendo disso, vamos configurar nosso serviço para que tenhamos hosts virtuais no servidor.

Configurando o Apache

O nosso primeiro passo é acessar o arquivo onde iremos configurar nossos hosts virtuais.

Esse arquivo, no nosso caso, fica em `/etc/httpd/conf/httpd.conf`.

Dentro do arquivo de configuração, vamos dizer que teremos um host virtual que atuará em qualquer endereço ip (*) e escutará na porta que é o padrão para websites (80), portanto, antes da última linha do arquivo, vamos escrever o seguinte trecho:

```
# Restante da configuração do Apache
<VirtualHost *:80>

</VirtualHost>
```

Também é preciso informar a pasta onde estarão os arquivos do nosso site. Vamos listar (ls) nossas pastas e saber quais temos disponíveis:

```
# ls
```

```
[root@srv01 www]# ls
acmetoon.com.br  voceanimado.com.br
[root@srv01 www]# _
```

Bem, temos dois diretórios, cada um deles tem como nome o domínio do site. Logo, vamos colocar os arquivos da `www.acmetoon.com.br`, na pasta que contém esse nome.

Esse tipo de configuração facilita na administração do servidor, pois sabemos que cada diretório pertence a um site e a qual site se trata. Facilitando a administração no futuro.

Temos que informar a pasta onde é a raiz de nossos documentos (DocumentRoot), isto é, onde se localizam nossos arquivos, seguido do caminho até o diretório:

```
<VirtualHost *:80>
    DocumentRoot /www/acmetoon.com.br
</VirtualHost>
```

Vamos dizer agora qual é o domínio do nosso site, ou seja, o nome do servidor (ServerName)

```
<VirtualHost *:80>
    DocumentRoot /www/acmetoon.com.br
    ServerName www.acmetoon.com.br
</VirtualHost>
```

Por fim, para que o Apache consiga ver os arquivos em nossa pasta, vamos dar a ele todas as permissões. Informaremos então nosso diretório e o caminho até ele (`<Directory "/www/acmetoon.com.br">`), em seguida, indicaremos que ele pode ler todos os arquivos (**Require all granted**):

```
<VirtualHost *:80>
    DocumentRoot /www/acmetoon.com.br
    ServerName www.acmetoon.com.br
    <Directory "/www/acmetoon.com.br">
        Require all granted
    </Directory>
</VirtualHost>
```

O site da AcmeToon está configurado. Vamos então atualizar as instruções do Apache para que ele reconheça nossas configurações. Para isso, vamos restartar o serviço, o comando **apachectl restart** faz isso para a gente:

```
[root@srv01 ~]# apachectl restart
[root@srv01 ~]# _
```

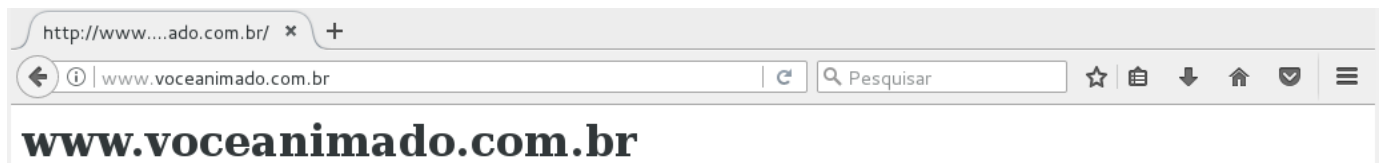
Será que funcionou a nossa configuração? Vamos abrir o navegador e verificar:



Boa! Está funcionando! Mas agora precisamos configurar nosso outro host. Podemos configurá-lo conforme fizemos no primeiro. A configuração é bem parecida, basta abrir a tag de virtual host e realizar as configurações que vimos a cima:

```
<VirtualHost *:80>
    DocumentRoot /www/voceanimado.com.br
    ServerName voceanimado.com.br
    <Directory "/www/voceanimado.com.br">
        Require all granted
    </Directory>
</VirtualHost>
```

Novamente, temos que reiniciar nosso serviço e testar a configuração do novo host:



Está funcionando como esperado. Já podemos hospedar nossos sites.

Configurando Subdomínios

Com o passar do tempo, a AcmeToon, decidiu criar um blog para a empresa. O blog faz parte do site da empresa e queríamos usar o domínio `acmetoon.com.br` para hospedá-lo também... Mas como?

Podemos configurar um [subdomínio](#), isto é, uma ramificação do site principal (`www.acmetoon.com.br`) que funcionará apenas para o blog da AcmeToon. Nossos hosts virtuais também podem ser utilizados para criar subdomínios.

A configuração seguirá o padrão das configurações anteriores, também teremos uma pasta para especificar esse website:

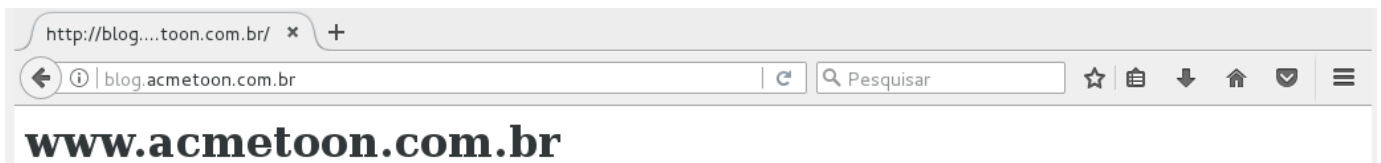
```
[root@srv01 www]# ls
acmetoon.com.br  blog.acmetoon.com.br  voceanimado.com.br
[root@srv01 www]# _
```

Assim os arquivos do blog ficarão separados dos demais.

A pasta onde é a raiz de nossos arquivos (DocumentRoot) será `blog.acmetoon.com.br`, também daremos ao Apache a permissão de acessá-la. A configuração completa fica assim:

```
<VirtualHost *:80>
    DocumentRoot /www/blog.acmetoon.com.br
    ServerName www.acmetoon.com.br
    <Directory "/www/blog.acmetoon.com.br">
        Require all granted
    </Directory>
</VirtualHost>
```


Vamos tentar acessar o blog agora e ver se está funcionando nossa configuração:



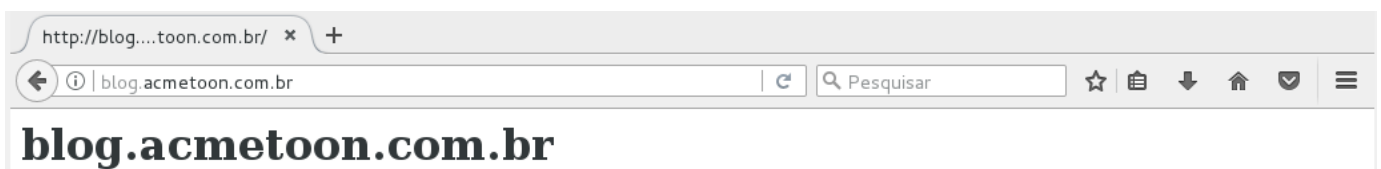
Hum... Por quê não apareceu a página do blog?

Como o blog é um subdomínio, temos que especificar isso na configuração do Apache, para que ele entenda e consiga acessá-lo. Ou seja, no local onde especificamos o nome do site, temos que dizer que queremos acessar o blog.

Para isso basta substituir o `www` por `blog` no `ServerName`:

```
<VirtualHost *:80>
    # Restante da configuração
    ServerName blog.acmetoon.com.br
    # Restante da configuração
</VirtualHost>
```

Vamos ver se agora seremos levados ao blog:



Oba! Conseguimos acessar nossa página do blog. Agora temos todos nossos sites funcionando no mesmo servidor.

Boas Práticas

Veja que fizemos a configuração dos hosts virtuais dentro do arquivo de configuração do Apache, ou seja, esse arquivo contém, além das configurações do próprio Apache, nossas especificações sobre os **hosts virtuais**.

Como não configuramos muitos hosts diferentes, isso pode não aparentar ser um problema, mas e se fossemos configurar mais cinco, dez ou mais hosts virtuais?

Da maneira como fizemos anteriormente, as especificações desses novos hosts virtuais estariam dentro do arquivo de configuração do Apache e isso aumentaria muito o tamanho do arquivo destinado às configurações do Apache.

Além do mais, imaginem buscar a configuração de um único host virtual quando existem vários em um mesmo local.

Pensando nisso, existe uma boa prática que consiste em criar arquivos separados para cada host virtual. Dessa forma, com códigos isolados uns dos outros, evitamos que o arquivo do Apache fique muito grande e difícil de ler, além de segmentar a configuração dos sites, o que deixa mais fácil a manutenção.

No post estamos mostrando uma configuração simples utilizando o próprio arquivo do Apache, apenas para mostrar a criação dos hosts virtuais, mas caso desejem ver a configuração utilizando essa boa prática, aqui está o [um exemplo](#) sobre como realizá-la.

Para saber mais

Como vimos, podemos ter quantos domínios e subdomínios quisermos em um único servidor. Quando isso ocorre ele recebe o nome de [servidor compartilhado](#), já que pode ser compartilhado com vários sites ao mesmo tempo.

Porém, dependendo do número de requisições que temos em nossos sites, isso pode causar um gargalo no processamento, já que tanto o processamento quanto a memória são compartilhado com os sites.

Por isso, nos dias de hoje grandes sites utilizam [servidores dedicados](#) e distribuídos e cada dia mais aderem ao advento da [nuvem](#) (**dados distribuídos entre servidores**), o que resolve muitos problemas.

Pontos de atenção

Estamos fazendo a configuração como superusuário, por isso **cuidado** para não executar um comando indevido e causar problemas no sistema.

Vale lembrar também que para nossos sites ficarem no ar, precisamos registrar os nomes em um [DNS](#).

Conclusão

Inicialmente estávamos com o problema de hospedar mais de um site no mesmo servidor. Vimos como podemos fazer isso utilizando hosts virtuais. Também, entendemos como é possível realizar a configuração de subdomínios dentro do servidor.

Além disso, conhecemos que uma possibilidade seria a virtualização do servidor do próprio servidor. Mas nesse caso ela não seria muito viável, já que teríamos que dar manutenção replicada em ambas as virtualizações.

Contudo, a virtualização é muito utilizada em hospedagem, devido a alguns benefícios de economia de energia, pelo fato de existir apenas uma máquina física.

E aí, o que achou? Conhecia essas técnicas? Conhece outras? Compartilhe conosco nos comentários.

Gosta da parte de servidores? Essa é só uma pequena parte do mundo de infraestrutura. Aqui na **Alura** temos uma [formação completa](#) em redes. Nela você aprenderá sobre protocolos de redes, **VLans**, segurança e muito mais!



Formação DevOps

Cursos de Infraestrutura

ARTIGOS DE TECNOLOGIA E NEGÓCIOS > INFRAESTRUTURA

PLANOS

TODOS OS CURSOS

GUIAS DE EMPRESAS

INSTRUTORES

COMO VIRAR INSTRUTOR

ARTIGOS