INSTALANDO UM SERVIDOR LAMP (Linux + Apache + MySQL + PHP)

É uma combinação de soluções destinada a sustentar um servidor com conteúdo Web e Banco de Dados.

INSTALANDO O APACHE

apt-get install apache2 apache2-utils ssl-cert

apache2 -> apache propriamente dito.

apache2-utils -> diversos utilitários de gerenciamento.

ssl-cert -> suporte a páginas seguras.

* Para reiniciar o servidor apache utilize:

/etc/init.d/apache2 restart

Acessando o endereço http://127.0.0.1, é possivel ver a página de boas vindas indicando que o servidor está funcionando.

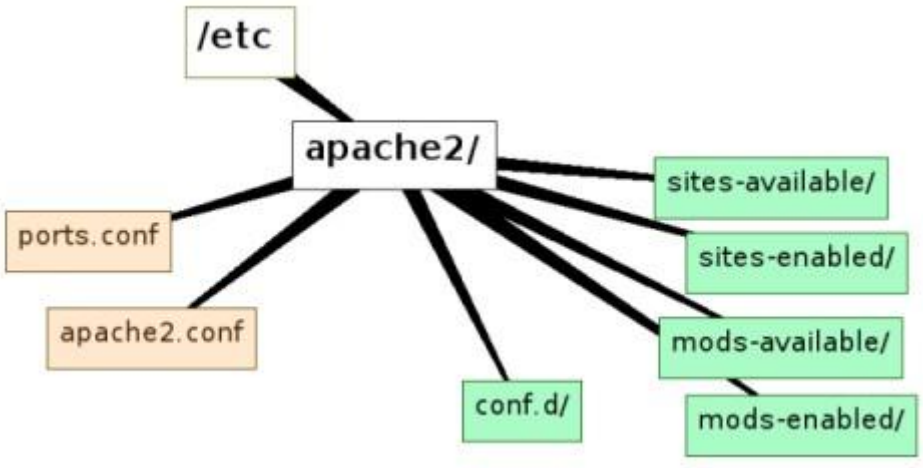
* Se não huover nenhum firewall ja pode ser acessado de qualquer host da rede.

** O diretório raiz do servidor web é /var/www.

O diretório raiz é definido através de uma opção dentro do arquivo principal de configuração /etc/apache2/default (a opção "DocumentRoot") e pode ser alterado caso desejado. Ao hospedar diversos sites no mesmo servidor, você define uma pasta raiz diferente para cada um.

ORGANIZAÇÃO DOS ARQUIVOS

A arquitetura modular do Apache é extendida também aos arquivos de configuração, tradicionalmente, a configuração do Apache é centralizada em um único arquivo, o "**httpd.conf**", que pode opcionalmente incluir referências a arquivos externos (includes) que permitem segmentar e organizar a configuração.



Todos os arquivos de configuração estão organizados dentro do diretório **"/etc/apache2"**. Dentro dele, temos:

"sites-available" e **"sites-enabled"** -> que contêm a configuração dos sites hospedados;
"mods-available" e "mods-enabled" -> que armazenam a configuração dos módulos;

o arquivo "ports.conf" -> onde vai a configuração das portas TCP que o servidor vai escutar;
o arquivo "apache2.conf" -> que armazena configurações diversas relacionadas ao funcionamento do servidor
e a pasta "conf.d" -> que armazena arquivos com configurações adicionais.

O Apache é capaz de hospedar simultaneamente vários sites, cada um representado por um arquivo de configuração diferente. Imagine o caso de uma empresa de hosting que mantém um servidor com 2.000 pequenos sites. Quando cada cliente registra seu site e assina o plano de hospedagem, você cria um novo arquivo dentro da pasta "sites-available" com as configurações necessárias e um link para ele na pasta "sites-enabled".
Como os nomes sugerem, a primeira pasta armazena a configuração de todos os sites (virtual hosts) hospedados no servidor, mas apenas os sites que estiverem presentes na pasta "sites-enabled" ficam disponíveis. Quando é necessário suspender temporariamente um site por falta de pagamento, por exemplo, você simplesmente remove o link na pasta "sites-enabled", sem precisar mexer na configuração.

Ao invés de criar e remover os links manualmente, você pode usar os comandos "**a2ensite**" e "**a2dissite**". Para ativar e desativar um site configurado no arquivo "/etc/apache2/sites-available/novo_site", por exemplo, os comandos seriam:

"/etc/apache2/sites-available/novo_site a2ensite novo_site (ativa)
a2dissite novo_site (desativa)

Quando o Apache é instalado, é criado por padrão o arquivo "/etc/apache2/sites-available/default", que contém a configuração de um site "raiz", que usa (por padrão) a pasta "/var/www" como diretório de páginas. Se o seu servidor web vai hospedar um único site, então essa configuração é suficiente, mas, caso você queira hospedar vários sites no mesmo servidor, é necessário criar uma pasta e um arquivo de configuração para cada site adicional.

Ex:
Seu servidor pode, por exemplo, hospedar o "joao.com.br" e o "maria.com.br". Um servidor DNS, mantido por você, é configurado para responder pelos dois domínios, em ambos os casos fornecendo o endereço IP do seu servidor web aos clientes. Na configuração do apache, criamos os arquivos "/etc/apache2/sites-available/joao" e "/etc/apache2/sites-available/maria", cada um configurado para utilizar uma pasta diferente. De acordo com sua preferência, podem ser usadas pastas dentro do diretório home de cada usuário, como em "/home/joao/html" e "/home/maria/html", ou subpastas dentro do diretório "/var/www", como em "/var/www/joao" e "/var/www/maria".

A mesma idéia se aplica aos módulos. A pasta "mods-available" contém a configuração e scripts de inicialização para todos os módulos disponíveis, mas apenas os módulos referenciados (através de um link) na pasta "mods-enabled" são realmente carregados.

Para desativar o suporte a PHP, por exemplo, você usaria o comando:
 # a2dismod php5
Para ativá-lo novamente, usaria:
 # a2enmod php5

Uma vez que um determinado módulo é ativado, ele fica automaticamente disponível para todos os sites hospedados no servidor.
Para que a alteração entre em vigor, é necessário reiniciar o serviço, usando o comando "/etc/init.d/apache2 force-reload" ou "/etc/init.d/apache2 restart" (os dois comandos fazem exatamente a mesma coisa):

 # /etc/init.d/apache2 force-reload

No arquivo "**ports.conf**", você pode alterar a porta padrão do seu servidor ou adicionar novas portas. Originalmente o arquivo vem com uma única linha:

Listen 80

Para fazer com que seu servidor escute também na porta 443 (a porta do HTTPS) e na porta 8080, por exemplo, você adicionaria duas novas linhas:

Listen 80
Listen 443
Listen 8080

INSTALANDO O SUPORTE AO PHP

O suporte a PHP é instalado através do pacote "**php5**". Para instalá-lo, basta usar o seguinte comando:

apt-get install php5

Com o interpretador PHP instalado, falta instalar o módulo do Apache 2, que está disponível através do pacote "libapache2-mod-php5":

apt-get install libapache2-mod-php5

O módulo "libapache2-mod-php5" é instalado dentro da pasta "**/usr/lib/apache2/modules/**" e os links são criados automaticamente ao instalar o pacote, mas você pode tirar qualquer dúvida usando o comando a2enmod:

 # a2enmod php5

Não esqueça de reiniciar o serviço para que o módulo seja carregado e a configuração entre em vigor:

 # /etc/init.d/apache2 force-reload ou **# service httpd restart**

Com o suporte a PHP ativado, o Apache continua exibindo diretamente páginas com extensão .htm ou .html, mas passa a entregar as páginas .php ou .phps ao interpretador php, que faz o processamento necessário e devolve uma página html simples ao Apache, que se encarrega de enviá-la ao cliente.

Quase sempre, os sistemas desenvolvidos em PHP utilizam também um banco de dados MySQL ou Postgre SQL.
Para que o interpretador PHP seja capaz de acessar o banco de dados, é necessário ter instalado (além do servidor MySQL propriamente dito) o módulo "php5-mysql", que faz a junção entre os dois componentes:

 # apt-get install php5-mysql

No caso do PostgreSQL, é utilizado o módulo "php5-pgsql", que tem a mesma função:

 # apt-get install php5-pgsql

Não se esqueça de reiniciar o Apache, para que as alterações entrem em vigor:

 # /etc/init.d/apache force-reload

Para verificar se o suporte a PHP está realmente ativo, crie um arquivo de texto chamado "info.php" (ou outro nome qualquer, seguido da extensão .php) dentro da pasta do servidor web, contendo apenas a linha abaixo:

<?php phpinfo( ); ?>

Salve o arquivo e abra a página através do navegador. A função "phpinfo", que usamos no arquivo, faz com que o servidor exiba uma página com detalhes da configuração do PHP e dos módulos ativos:
Depois de verificar, remova o arquivo, pois não é interessante que essas informações fiquem disponíveis ao público.

INSTALE O MySQL

O primeiro passo é instalar o servidor MySQL propriamente dito:

apt-get install mysql-server

Antes de iniciar o serviço, rode o comando "mysql_install_db". Ele prepara o terreno, criando a base de dados "mysql" (usada para armazenar a configuração do servidor MySQL, incluindo informações sobre os usuários e sobre as demais bases de dados) e também uma base de dados chamada "test", que pode ser usada para testar o servidor:

 # mysql_install_db

O passo seguinte é ativar o servidor MySQL:

 # /etc/init.d/mysql start

O MySQL possui um usuário padrão chamado "root", que, assim como o root do sistema, tem acesso completo a todas as bases de dados e é usado para fazer a configuração inicial do sistema, assim como tarefas de manutenção.

Se você precisar trocar a senha posteriormente utilize:

 # mysqladmin -u root -p password 12345

Enter password: ********

Veja que nesse caso é necessário incluir a senha antiga ao executar o comando, antes de continuar, já que do contrário teríamos uma brecha óbvia de segurança.

Depois de definir a senha, o próximo passo é criar uma base de dados.

Para acessar o MySQL, use o comando:

 # mysql -u root -p <enter>

Enter password: <senha>

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 43 to server version: 4.0.15-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>

Veja que o cabeçalho normal do bash foi substituído por um "mysql>", que lembra onde você está. Para sair, pressione "Ctrl+C" ou execute o comando "quit".

Dentro do prompt do MySQL, use o comando "CREATE DATABASE" (criar base de dados), seguido pelo nome desejado.

Ex:

Um detalhe importante é que todos os comandos dados dentro do prompt do MySQL devem terminar com ponto-e-vírgula:

mysql> **CREATE DATABASE novo**;
Query OK, 1 row affected (0.04 sec)

Para confirmar, use o comando "SHOW DATABASES", que lista as bases de dados criadas no servidor, como em:

mysql> **SHOW DATABASES;**

```
+--------------------------------+
|Database                        |
+--------------------------------+
|Information_schema              |
|mysql                           |
|novo                            |
|test                            |
+--------------------------------+
```

Note que além da base "novo" que criamos, existem mais bases de dados, criadas durante a instalação. Estas bases são para uso interno do MySQL, incluindo o armazenamento das, enquanto a base "test" é uma DB vazia, que pode ser usada para fins de teste.

INSTALANDO phpMyAdmin

Depois dessa configuração inicial, você pode experimentar instalar um gerenciador gráfico para facilitar a manutenção do seu servidor MySQL. Uma boa opção neste caso é o **phpMyAdmin**.

Para instalá-lo, basta instalar o pacote "phpmyadmin", como em:

 # apt-get install phpmyadmin

O phpMyAdmin é um gestor de configuração escrito em PHP que trabalha em conjunto com o Apache. Ele permite que você crie bases de dados, ajuste as permissões de acesso dos usuários, faça backup, e diversas outras atividades administrativas de uma forma mais simples que através do prompt de comando.

Uma vez instalado, ele pode ser acessado através do endereço "http://servidor/phpmyadmin/". Na tela inicial, você pode se logar usando qualquer uma das contas registradas no MySQL. Use o root para tarefas administrativas, quando for necessário ter acesso a todas as bases ou fazer backup de tudo, e uma das contas restritas para acessar uma base específica:

Instalando:

Selecione a opção "apache2" quando o script perguntar sobre o servidor web usado e responda "sim" quando ele perguntar se você deseja reiniciar o serviço, cadastre as senhas necessárias e faça um teste de acesso.

Ex:
http://192.168.0.10/phpmyadmin
Usuário: root
Senha: 12345