Atividade-5

Instalar dois domínios em um mesmo Servidor

Para hospedar dois domínios diferentes (por exemplo, www.site1.com e www.site2.com) no mesmo servidor Apache2 e configurá-los para usar HTTPS, você precisa seguir algumas etapas. Essencialmente, você configurará hospedagem virtual baseada em nome para seus domínios e, em seguida, protegerá esses domínios com certificados SSL.

Instalar o Apache2. Se você ainda não instalou o Apache2, pode fazê-lo usando o gerenciador de pacotes da sua distribuição Linux. No Debian/Ubuntu, você usaria:

```
sudo apt update
sudo apt install apache2
```

Configurar Hospedagem Virtual: Criar Diretórios para os Sites: Primeiro, é uma boa prática criar um diretório para cada um dos seus sites no diretório /var/www/. Isso ajuda a manter os arquivos de cada site organizados.

```
sudo mkdir -p /var/www/site1.com/public_html
sudo mkdir -p /var/www/site2.com/public_html
```

Definir Permissões: Defina as permissões adequadas para esses diretórios.

```
sudo chown -R $USER:$USER /var/www/site1.com/public_html
sudo chown -R $USER:$USER /var/www/site2.com/public_html
```

Criar Arquivos de index.html de Teste: Para testar, crie um arquivo index.html simples em cada diretório.

```
/var/www/site1.com/public_html/index.html (Coloque um conteúdo de sua preferencia)
/var/www/site2.com/public_html/index.html  (Coloque um conteúdo de sua preferencia)
```

Criar Arquivos de Configuração de Hospedagem Virtual: Para cada domínio, crie um arquivo de configuração dentro de /etc/apache2/sites-available/. Você pode usar o editor de texto de sua preferência.

Para site1.com:

```
sudo nano /etc/apache2/sites-available/site1.com.conf
```

Adicione o seguinte conteúdo, ajustando para o seu domínio:

```
<VirtualHost *:80>
        ServerAdmin webmaster@site1.com
        ServerName site1.com
        ServerAlias www.site1.com
        DocumentRoot /var/www/site1.com/public_html
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Para site2.com:

sudo nano /etc/apache2/sites-available/site2.com.conf

Adicione o seguinte conteúdo, ajustando para o seu domínio:

```
<VirtualHost *:80>
        ServerAdmin webmaster@site2.com
        ServerName site2.com
        ServerAlias www.site2.com
        DocumentRoot /var/www/site2.com/public_html
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Ativar os Sites: Use o comando a2ensite para ativar cada configuração de site.

```
sudo a2ensite site1.com.conf
sudo a2ensite site2.com.conf
```

Recarregar o Apache2: Para aplicar as mudanças, recarregue o serviço Apache2.

```
sudo systemctl reload apache2
```

Ativando o DNS

Para que os nomes de domínio www.site1.com e www.site2.com sejam resolvidos para o endereço IP do seu servidor Apache2 na Internet, é necessário configurar um sistema de nomes de domínio (DNS). O BIND (Berkeley Internet Name Domain) é um dos servidores DNS mais usados e pode ser configurado para esse propósito.

Instalar o BIND: instale o BIND no seu servidor.

```
sudo apt install bind9
```

Configurar Zonas DNS para Seus Domínios

Editar o Arquivo de Configuração do BIND: Abra o arquivo de configuração principal do BIND, named.conf.local, para adicionar suas zonas DNS.

```
sudo nano /etc/bind/named.conf.local
```

Adicionar Zonas: No arquivo, adicione uma seção de zona para cada um dos seus domínios. Isso dirá ao BIND onde encontrar os arquivos de configuração para cada domínio.

```
zone "site1.com" {
    type master;
    file "/etc/bind/zones/site1.com.db"; // Caminho para o arquivo de dados da zona
};
```

```
zone "site2.com" {
    type master;
    file "/etc/bind/zones/site2.com.db"; // Caminho para o arquivo de dados da zona
};
```

Criar Diretório para os Arquivos de Zona: É uma boa prática manter os arquivos de zona em um diretório separado.

```
sudo mkdir /etc/bind/zones
```

Criar Arquivos de Zona: Para cada domínio, crie um arquivo de zona no diretório especificado. Esses arquivos conterão os registros DNS, incluindo o registro A que aponta para o endereço IP do seu servidor Apache2.

Para site1.com:

```
sudo nano /etc/bind/zones/site1.com.db
```

Adicione o seguinte conteúdo, ajustando o endereço IP (192.168.x.x) para o do seu servidor:

```
$TTL    604800
@       IN      SOA     ns1.site1.com. admin.site1.com. (
                        2       ; Serial
                        604800        ; Refresh
                        86400       ; Retry
                        2419200         ; Expire
                        604800 )      ; Negative Cache TTL
; nameservers
@       IN      NS      ns1.site1.com.
ns1     IN      A       192.168.x.x

; A records for web server
@       IN      A       192.168.x.x
www     IN      A       192.168.x.x
```

Repita para site2.com, ajustando conforme necessário.

```
sudo nano /etc/bind/zones/site2.com.db
```

Adicione o seguinte conteúdo, ajustando o endereço IP (192.168.x.x) para o do seu servidor:

```
$TTL    604800
@       IN      SOA     ns1.site2.com. admin.site2.com. (
                        2       ; Serial
                        604800        ; Refresh
                        86400       ; Retry
                        2419200        ; Expire
                        604800 )      ; Negative Cache TTL
; nameservers
@       IN      NS      ns1.site2.com.
ns1     IN      A       192.168.x.x

; A records for web server
@       IN      A       192.168.x.x
www     IN      A       192.168.x.x
```

Verificar e Reiniciar o BIND

Verificar a Configuração: Antes de reiniciar o BIND, é uma boa prática verificar se há erros na sua configuração.

```
sudo named-checkconf

sudo named-checkzone site1.com /etc/bind/zones/site1.com.db

sudo named-checkzone site2.com /etc/bind/zones/site2.com.db
```

Reiniciar o BIND: Se não houver erros, reinicie o serviço BIND.

```
sudo systemctl restart bind9
```

Ativar o SSL Autoasinado - HTTPS

Gerar um Certificado SSL Autoassinado

Criar uma Chave Privada e um Certificado: No terminal, use o OpenSSL para gerar sua chave privada e certificado autoassinado. O exemplo abaixo cria uma chave RSA de 2048 bits e um certificado válido por 365 dias.

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/site1.com.key -out /etc/ssl/certs/site1.com.crt
```

Você será solicitado a inserir informações que serão incorporadas ao seu certificado.

Para site2.com, basta repetir o comando, ajustando os nomes dos arquivos de chave e certificado.

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/site2.com.key -out /etc/ssl/certs/site2.com.crt
```

Configurar o Apache para Usar o SSL: Para cada domínio, você precisa configurar o Apache para usar a chave privada e o certificado que você acabou de criar. Isso é feito modificando ou criando um novo arquivo de configuração de hospedagem virtual para cada site na pasta /etc/apache2/sites-available/.

Para site1.com, crie ou edite o arquivo de configuração:

```
sudo nano /etc/apache2/sites-available/site1.com-ssl.conf
```

Adicione o seguinte conteúdo, ajustando os caminhos para sua chave e certificado:

```
<VirtualHost *:443>
    ServerAdmin webmaster@site1.com
    ServerName site1.com
    ServerAlias www.site1.com
    DocumentRoot /var/www/site1.com/public_html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/site1.com.crt
    SSLCertificateKeyFile /etc/ssl/private/site1.com.key
</VirtualHost>
```

Repita o processo para site2.com, garantindo que você use os arquivos de certificado e chave corretos para esse domínio.

```
sudo nano /etc/apache2/sites-available/site2.com-ssl.conf
```

Adicione o seguinte conteúdo, ajustando os caminhos para sua chave e certificado:

```
<VirtualHost *:443>
    ServerAdmin webmaster@site2.com
    ServerName site2.com
    ServerAlias www.site2.com
    DocumentRoot /var/www/site2.com/public_html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/site2.com.crt
    SSLCertificateKeyFile /etc/ssl/private/site2.com.key
</VirtualHost>
```

Ativar o Módulo SSL e os Sites

Ativar o Módulo SSL do Apache: Se ainda não estiver ativado, você precisa ativar o módulo SSL do Apache.

```
sudo a2enmod ssl
```

Ativar os Sites: Ative os sites que você configurou para usar SSL.

```
sudo a2ensite site1.com-ssl.conf
sudo a2ensite site2.com-ssl.conf
```

Recarregar o Apache: Para aplicar as mudanças, recarregue o Apache.

```
sudo systemctl reload apache2
```

Testar a Configuração

Testar com um Navegador ou Ferramenta de Linha de Comando: Você pode testar se o SSL está funcionando corretamente acessando seus sites com https:// na frente do domínio. Navegadores web alertarão sobre o certificado autoassinado, mas você deve ser capaz de prosseguir após uma exceção de segurança.

Verificar a Configuração do SSL: Utilize ferramentas como o OpenSSL para verificar se o SSL está corretamente configurado.

```
openssl s_client -connect site1.com:443
openssl s_client -connect site2.com:443
```

Lembre-se de que certificados autoassinados podem gerar avisos de segurança nos navegadores dos usuários, pois não são emitidos por uma Autoridade Certificadora (AC) reconhecida. Eles são melhores para testes ou ambientes internos onde a confiança dos usuários já é estabelecida por outros meios.