



**CARLOS MAGNO DO NASCIMENTO JUNIOR - 10A**  
**GRAZYELLE RODRIGUES DE FARIA - 22A**  
**LUIS FELIPE BAPTISTA SERAPIAO - 10A**

**RELATÓRIO TRABALHO DE INSTALAÇÃO DE  
SERVIÇOS DE REDE II**

**Lavras - MG**  
**2022**

## 1. INTRODUÇÃO E OBJETIVO

O presente trabalho tem por objetivo a expansão do trabalho realizado anteriormente na primeira etapa, adicionando servidor FTP na máquina 192.168.1.42, que atualmente já hospeda nosso servidor WEB, e servidor DNS na máquina 192.168.1.43.

## 2. SERVIDOR FTP

### a. Instalando e configurando UFW

Para que possamos abrir portas para nosso servidor ftp, é antes necessário fazer a permissão de uso delas no firewall do sistema, para isto, precisamos instalar “ufw” utilizando comando:

```
$ sudo apt install ufw
```

Com ufw instalado, usando permissão sudo, podemos agora começar a adicionar permissões no nosso firewall em sua configuração, as seguintes permissões foram adicionadas: OpenSSH, 20/tcp, 21/tcp, 990/tcp e 40000:50000/tcp.

Todas adicionadas utilizando o comando:

```
$ sudo ufw allow <item_a_ser_permitido>
```

Seus status e permissões podem ser observados no nosso servidor ftp utilizando o seguinte comando:

```
$ sudo ufw status
```

```
[10:37:39] DEBIAN: aluno@debian [~]$ sudo ufw status
[sudo] senha para aluno:
Status: active

To Action From
--
20/tcp ALLOW Anywhere
21/tcp ALLOW Anywhere
OpenSSH ALLOW Anywhere
990/tcp ALLOW Anywhere
40000:50000/tcp ALLOW Anywhere
20/tcp (v6) ALLOW Anywhere (v6)
21/tcp (v6) ALLOW Anywhere (v6)
OpenSSH (v6) ALLOW Anywhere (v6)
990/tcp (v6) ALLOW Anywhere (v6)
40000:50000/tcp (v6) ALLOW Anywhere (v6)
```

**Figura:** Permissões adicionadas a ufw.

Houve um pouco de confusão nesta etapa, pois inicialmente, havia iniciado a a configuração do vsftpd antes, e todos os guias seguidos exigiam comandos utilizando uwp, mas a vm não respondia por eles. Também por ter seguido 2 guias de instalação de vsftpd

### **b. Instalando e configurando vsftpd**

Com ufw instalado, agora podemos voltar a instalação do nosso serviço ftp, o serviço escolhido foi vsftpd pela ampla quantidade de material encontrado para instalação e configuração do mesmo em uma máquina Debian (ambos para nosso fortúnio quanto confusão).

Iniciamos instalando nosso serviço na máquina com:

**\$ sudo apt install vsftpd**

E verificamos se foi instalado corretamente checando sua versão:

**\$ vsftpd -versions**

Agora é a etapa que é feita a configuração das portas utilizando uwp, explicado na sessão anterior. Esta etapa foi seguida a partir de múltiplas fontes, referenciadas no fim do documento, assim como foi a configuração do servidor vsftpd, por isso, é muito provável que há “bloat” de configurações e permissões.

Para que não haja o risco de que tenhamos problemas demais para resolver caso cometessemos algum erro na configuração do vsftpd, foi criado uma cópia do seu arquivo de configuração original, utilizando o comando:

**\$ sudo cp/etc/vsftpd.conf /etc/vsftpd.conf.orig**

Com o backup do arquivo feito, podemos iniciar a configurar nosso servidor, acessaremos o arquivo de configuração utilizando nano, utilizando:

**\$ sudo nano/etc/vsftpd.conf**

Adicionamos as seguintes configurações no fim de nosso arquivo:

```
listen=NO
listen_ipv6=YES
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
chroot_local_user=YES
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=Yes
pasv_enable=Yes
pasv_min_port=10000
pasv_max_port=10100
allow_writeable_chroot=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
userlist_enable=YES
userlist_file=/etc/vsftpd.userlist
userlist_deny=NO
force_local_logins_ssl=NO
force_local_data_ssl=NO
```

Por fim, é necessário que adicionemos um usuário com permissão de acesso ao nosso servidor ftp, para esta etapa, decidimos manter o usuário padrão do sistema como nosso usuário de acesso ftp, ou seja, usuário “aluno” com senha “grupou”.

Usuário pode ser adicionado a lista de usuários do vsftpd usando o comando:

```
$ echo "aluno"|sudotee-a/etc/vsftpd.userlist
```

Com nossa userlist autorizada no nosso arquivo de configurações feita anteriormente, podemos reiniciar nosso servidor utilizando:

```
$ sudo systemctl restart vsftpd
```

Assim como habilitando nosso servidor para uso, assim para iniciar no startup da máquina:

```
$ sudo systemctl enable vsftpd
```

Por fim, também reiniciamos a vm:

## \$ sudo reboot

Checando status do serviço:

```
{14:17:10} DEBIAN: aluno@debian [~]$ sudo systemctl status vsftpd
[sudo] senha para aluno:
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset:
   Active: active (running) since Mon 2022-09-12 21:32:08 -03; 5 days ago
     Main PID: 630 (vsftpd)
       Tasks: 1 (limit: 1133)
      Memory: 2.2M
         CPU: 142ms
        CGroup: /system.slice/vsftpd.service
               └─630 /usr/sbin/vsftpd /etc/vsftpd.conf
```

Agora temos um servidor ftp funcional em nossa máquina usando vsftpd!

Precisamos apenas instalar os comandos ftp para acessá-lo:

## \$ sudo apt-get install ftp

Teste:

```
luiz@luiz-Aspire-A315-41:~$ ftp 192.168.1.42
Connected to 192.168.1.42.
220 (vsFTPd 3.0.3)
Name (192.168.1.42:luiz): aluno
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put testeftp.txt
local: testeftp.txt remote: testeftp.txt
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
9 bytes sent in 0.00 secs (25.4755 kB/s)
ftp> get testeftp.txt
local: testeftp.txt remote: testeftp.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for testeftp.txt (9 bytes).
226 Transfer complete.
9 bytes received in 0.00 secs (11.4292 kB/s)
ftp> █
```

**Figura:** teste de download e upload via ftp

### 3. SERVIDOR DNS

Para nosso servidor DNS, fizemos sua configuração utilizando **Bind9**, existe amplo material envolvendo essa ferramenta e por isso decidimos utiliza-la na nossa vm.

Iniciamos fazendo a instalação do Bin9, assim como dnsutils, com:

```
$ sudo apt install bind9 dnsutils
```

Em seguida, a fim de atender o requisito de nossas vms devem passar por nosso servidor dns, devemos alterar para que nosso servidor consulte em si próprio, como pedido nos requisitos, para isso, acessamos /etc/resolv.conf, criamos o backup deste arquivo no mesmo diretório como resolv.conf.old e o editamos com nano.

Dentro do arquivo, adicionamos que queremos fazer buscas no nosso localhost, da seguinte forma:

```
nameserver 192.168.1.43  
Search localhost
```

Depois disso, foi feita a configuração do arquivo /etc/bind/named.conf.options, neste arquivo fizemos a configuração da nossa acl (lista de controle de acesso), adicionaremos ele no início do nosso arquivo, assim:

```
acl internals{  
    ::1/128;  
    192.168.1.0/24; // nossa rede  
};
```

Agora dentro de options, adicionamos algumas configurações, inicialmente, nossos forwarders, para que possamos encaminhar consultas de serviços externos ao nosso servidor, usaremos dns da cloudflare e do google:

```
forwarders {  
    // Cloudflare Public DNS (ipv4)  
    1.1.1.1;  
    1.0.0.1;  
    // Cloudflare Public DNS (ipv6)  
    2606:4700:4700::1111;  
    2606:4700:4700::1001;  
  
    // Google Public DNS (ipv4)  
    8.8.8.8;
```

```
8.8.4.4;  
//Google Public DNS (ipv6)  
2001:4860:4860::8888;  
2001:4860:4860::8844;  
};
```

Por fim, faremos algumas configurações de segurança, incluindo limitar nosso servidor a apenas ouvir pedidos de consultas das nossas vms (do próprio servidor e nossa máquina web/ftp 192.168.1.42):

```
// Segurança:  
listen-on { 192.168.1.43; 192.168.1.42; }; // escuta as vms  
listen-on-v6 {::1};  
  
allow-query {internals};  
allow-recursion {internals};  
allow-transfer {none};  
version none;
```

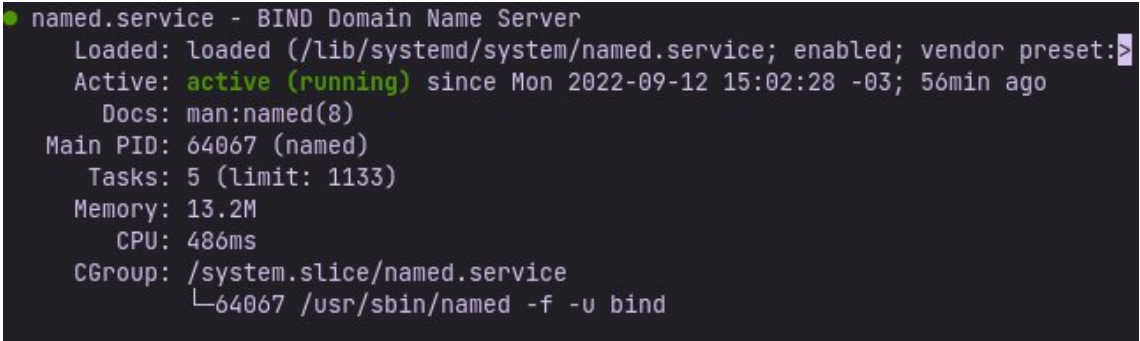
Finalizado a configuração, fizemos uma checkagem no arquivo com **named-checkconf** para ver se apontava erros na nossa configuração.


Terminada essas configurações, nosso servidor estava pronto para seus testes iniciais, para isso reiniciamos ele com:

```
$ systemctl restart bind9
```

E verificamos seu status com:

```
$ systemctl status bind9
```



```
● named.service - BIND Domain Name Server  
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: )  
   Active: active (running) since Mon 2022-09-12 15:02:28 -03; 56min ago  
     Docs: man:named(8)  
  Main PID: 64067 (named)  
    Tasks: 5 (limit: 1133)  
  Memory: 13.2M  
     CPU: 486ms  
   CGroup: /system.slice/named.service  
           └─64067 /usr/sbin/named -f -u bind
```

Por fim, foi feito teste se nosso servidor consegue resolver nomes externos a ele, no caso, utilizamos ufla.br:

```
[18:33:00] DEBIAN: aluno@debian [/etc/bind]$ nslookup ufla.br
Server:          192.168.1.43
Address:         192.168.1.43#53

Non-authoritative answer:
Name:   ufla.br
Address: 200.131.250.18
Name:   ufla.br
Address: 2801:a6::250:56ff:fe9f:8
```

Como pode ser visto, o endereço foi encontrado, e o servidor utilizado na consulta foi nosso servidor configurado como 192.168.1.43.

Apesar de nosso servidor estar “pronto”, ainda nos faltava adicionar endereços locais dos nossos servidores web e ftp, assim como configurar nossa outra máquina para que suas consultas fossem feitas através de nosso servidor.

Foi criada uma nova “zona” no arquivo **named.conf.default-zones**, com o nome escolhido “**grupou.gcc125.br**”, da seguinte forma:

```
zone "grupou.gcc125.br"{
    type master;
    file "/etc/bind/db.grupou.gcc125.br"
};
```

Após isso, foi criado o arquivo declarado na nossa zona, fazendo uma cópia do arquivo db.empty com o mesmo nome declarado na nossa zona “db.grupou.gcc125.br”. Neste arquivo, foi adicionado a seguinte configuração:

```
$TTL 86400
@      IN      SOA    ns1.grupou.gcc125.br. root.grupou.gcc125.br. (
                        3          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        86400 )    ; Negative Cache TTL
;
@      IN      NS     ns1
ns1    IN      A      127.0.0.1
www    IN      A      192.168.1.42
ftp    IN      A      192.168.1.42
```

Agora na nossa zona “grupou.gcc125.br” foi criado 3 endereços, nosso servidor dns ns1, web www e por fim ftp ftp.



Alguns testes:

```
[17:23:37] DEBIAN: aluno@debian [/etc/bind]$ ftp ftp.grupou.gcc125.br
Connected to ftp.grupou.gcc125.br.
220 (vsFTPd 3.0.3)
Name (ftp.grupou.gcc125.br:aluno): aluno
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ^C
ftp> bye
221 Goodbye.
```

**Figura:** teste de acesso a servidor ftp via endereço criado.

```
[17:31:01] DEBIAN: aluno@debian [/etc/bind]$ ping www.grupou.gcc125.br
PING www.grupou.gcc125.br (192.168.1.42) 56(84) bytes of data.
64 bytes from 192.168.1.42 (192.168.1.42): icmp_seq=1 ttl=64 time=0.364 ms
64 bytes from 192.168.1.42 (192.168.1.42): icmp_seq=2 ttl=64 time=0.211 ms
64 bytes from 192.168.1.42 (192.168.1.42): icmp_seq=3 ttl=64 time=0.185 ms
64 bytes from 192.168.1.42 (192.168.1.42): icmp_seq=4 ttl=64 time=0.208 ms
64 bytes from 192.168.1.42 (192.168.1.42): icmp_seq=5 ttl=64 time=0.260 ms
^C
--- www.grupou.gcc125.br ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4080ms
rtt min/avg/max/mdev = 0.185/0.245/0.364/0.064 ms
```

**Figura:** teste de ping a nosso servidor web via endereço criado.

```
[18:37:03] DEBIAN: aluno@debian [/etc/bind]$ nslookup ns1.grupou.gcc125.br
Server:          192.168.1.43
Address:         192.168.1.43#53

Name:   ns1.grupou.gcc125.br
Address: 192.168.1.43
```

**Figura:** busca de registro dns no nosso endereço ns1.

Agora, falta adicionar nosso servidor dns a nossa vm 192.168.1.42:

Pesquisando, foi descoberto que apenas atualizar nosso resolv.conf da segunda máquina, podia não resolver ou até voltar a configuração inicial e apenas trocado o endereço adicionado do nosso servidor dns, por isso, foi instalado nela o serviço resolveconf.service, com ele, podemos configurar o arquivo /etc/resolvconf/resolv.conf.d/head, que manterá configurações feitas, mesmo após possíveis reboots, a configuração feita no arquivo foi:

```

GNU nano 3.4 /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
# 127.0.0.53 is the systemd-resolved stub resolver.
# run "resolvectl status" to see details about the actual nameservers.

nameserver 192.168.1.43
search ns1.grupou.gcc125.br

```

E por fim, para testar nossa configuração, fizemos um nslookup na máquina 42, do endereço ns1, e endereço foi encontrado. (ip addr usado anteriormente para demonstrar endereço da máquina que rodou o comando).

```

[18:44:45] DEBIAN: aluno@debian [~]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:46:a3:12 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 192.168.1.42/24 brd 192.168.1.255 scope global ens192
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe46:a312/64 scope link
        valid_lft forever preferred_lft forever
[18:45:20] DEBIAN: aluno@debian [~]$ nslookup ns1.grupou.gcc125.br
Server:      192.168.1.43
Address:     192.168.1.43#53

Name:   ns1.grupou.gcc125.br
Address: 192.168.1.43
[18:45:29] DEBIAN: aluno@debian [~]$

```

Endereço google.com, teste de forwarding:

```
[14:27:57] DEBIAN: aluno@debian [~]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:46:a3:12 brd ff:ff:ff:ff:ff:ff
    altnam enp11s0
    inet 192.168.1.42/24 brd 192.168.1.255 scope global ens192
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe46:a312/64 scope link
        valid_lft forever preferred_lft forever
[14:28:11] DEBIAN: aluno@debian [~]$ nslookup google.com
Server:      192.168.1.43
Address:     192.168.1.43#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.79.14
Name:   google.com
Address: 2800:3f0:4004:80a::200e
```

#### 4. PROBLEMAS ENFRENTADOS

Vários problemas foram encontrados na realização desta atividade, um servidor uwp não configurado, configuração com bastante bloat do servidor ftp, vinda de debugs de várias fontes diferentes, dificuldades para que servidor ftp funcionasse sem encrypting.

Mas servidor DNS foi o maior causador de dor de cabeças, a extensa configuração envolvendo vários arquivos diferentes, assim como novamente, vários fontes diferentes consultadas, geraram boa dor de cabeça. Erros de syntaxe causaram fortes problemas também, assim como uma configuração inicial usando endereço 127.0.0.1 a qual não conseguimos utilizar na outra VM, necessitando reconfigurar o endereço do servidor.

Trocar o servidor DNS da segunda máquina para nosso servidor configurado foi mais desafiador do que esperado, e necessitou de instalação de ferramentas externas para manter configuração (ou pelo menos, é o que achamos que ajudou).

É provável que a necessidade de reboots do serviço dns para cada mudança tenham feito a impressão que haviam mais problemas que de fato existiam.

Ao fazer primeiro upload desse relatório, também foi descoberto que servidor web apache quebrou ao longo do processo de configuração de um dos serviços (provavelmente o ftp/uwp), foi resolvido desinstalando e reinstalando serviço, seguido de repetir paços feitos de configuração do nosso primeiro relatório de trabalho com vms.

## 5. RESULTADOS E CONCLUSÕES

Apesar das grandes dificuldades enfrentadas pelo grupo para cumprir os requisitos pedidos pelo professor, conseguimos com sucesso criar nossos servidores, com acesso ftp ao usuário “aluno”, na mesma máquina do servidor web, assim como um servidor DNS na outra máquina, permitindo que as duas vms façam consultas dns no nosso servidor. Também conseguimos adicionar endereços com nomes propícios para nossos serviços, testa-los e incluí-los neste relatório.

## 6. REFERÊNCIAS

DIFITALOCEAN, [Initial Server Setup with Debian 10](#) – **configuração e instalação uwp**  
LINUXHINT, [How to Setup vsftpd ftp server on debian 10?](#) - **configuração e instalação vsftpd**

DIGITALOCEAN, [How To Set Up vsftpd for a User's Directory on Debian 10](#) -  
**configuração e instalação vsftpd**

YOUTUBE, Prof. Juliano Ramos, [Aula 4: Curso Servidor Debian - Instalando um servidor DNS](#) – **instalação e configuração de servidor dns bind9**

TECMINT, [How To Set Permanent DNS Nameservers in Ubuntu and Debian](#) –  
**instalação e configuração do resolvconf.service**

YOUTUBE, Fernando Leonid, [01 - Servidor DNS BIND9 \[ Debian 10 \]](#) – **configuração de zonas**

DIGITALOCEAN, [How To Configure BIND as a Private Network DNS Server on Debian 9](#) – **instalação e configuração de server dns, configuração de zonas**