

# Instalação e Configuração de Serviços

Prof. Thiago Guimarães Tavares - [thiagogmta@gmail.com](mailto:thiagogmta@gmail.com) - IFTO Campus Palmas

# Saudações

Este livro foi escrito com o objetivo de servir de base para o estudo e prática de configuração de servidores Linux. Observando o impasse da necessidade de infraestrutura física para o estudo da disciplina de Redes e suas derivações, como a configuração de Servidores, algumas ferramentas foram projetadas de forma a simular o ambiente físico. Uma dessas ferramentas é o **Netkit**. Essa ferramenta permite criarmos redes virtuais de computadores com a virtualização de hosts que consomem pouquíssimos recursos do computador hospedeiro. Dessa forma podemos não só simular diversas topologias de redes como também realizar a configuração de serviços e servidores linux.

Você vai encontrar nesse material um guia prático que irá lhe nortear em relação a configuração de alguns serviços em servidores linux utilizando como base o ambiente de testes Netkit. Apesar do processo ser realizado em um ambiente de testes o processo de instalação e configuração dos serviços utilizando o Netkit como meio ocorre tal como seria em um computador real. Justifica-se a utilização da ferramenta:

- Possibilidade de configuração de serviços em um ambiente de testes controlado.
- Baixo investimento em infraestrutura (nesse contexto apenas um computador real é necessário para simular uma rede complexa com computadores, roteadores e servidores).
- Diversidade de possibilidades de testes.

## Como este livro está organizado

O primeiro capítulo do livro aborda a ferramenta Netkit que será utilizada. Nesse capítulo será apresentada a instalação, configuração e utilização básica da ferramenta como: Criação de uma rede ponto a ponto; Configuração de um laboratório de redes; Compartilhamento de acesso a internet entre um host e seus clientes; Compartilhamento de arquivos entre o host anfitrião e seus hospedeiros; Modificação do pacote de aplicativos que compõem o Netkit. Os demais capítulos abordam a instalação e configuração dos respectivos serviços: DHCP; DNS; DDNS; Apache; FTP; Samba; Proxy e Firewall.

**A cada nova abordagem este guia encontra-se organizado da seguinte maneira:**

### 1. Apresentação

- Abordagem geral sobre o assunto respondendo inicialmente três questionamentos:
  - O que é?
  - Para que serve?
  - Como Funciona?

### 2. Exemplificação

- Redação do processo de instalação e configuração do serviço em questão. Essa sessão está organizada em capítulos que variam em quantidade e extensão de acordo com o tópico abordado.

### 3. Problematização

- Apresentação de alguma problematização a ser resolvida com objetivo na aplicação e fixação do conhecimento obtido. As problematizações estão inseridos ao final de capítulos chaves durante a exemplificação.

## Caixas de diálogo

No decorrer deste livro você poderá encontrar caixas de diálogo precedidas de ícones específicos. Confira o significado de cada uma delas:



Essa marcação representa algum complemento de informação. Chama atenção para alguma informação auxiliar.



Essa marcação apresenta alguma dica!



Essa marcação é utilizada para chamar atenção a uma informação importante para que não se esqueça.



Essa marcação é um aviso para que tenha cautela ou se atente para algo.



Essa marcação lhe chama atenção para que tenha cuidado, provavelmente para que se assegure de algo.

## Códigos e comandos

Sempre que for necessário a execução de um comando no terminal esses comando serão apresentados em destaque e precedidos por \$. Sempre que necessário, as linhas de comando serão marcadas por números e a respectiva descrição daquela instrução como exibido a seguir:

```
$ mkdir pasta ①  
$ cd pasta ②
```

① Comando para criar um diretório

② Comando para acessar o diretório criado

## Como sugiro que você utilize este livro

- Procure compreender a definição abordada na **Apresentação** de cada capítulo. Inevitavelmente é frustrante realizar alguma tarefa sem compreender qual seu objetivo.
- Leia atentamente todo o conteúdo apresentado na **Exemplificação** e só após tente, passo a passo, realizar as configurações. Algumas implementações exigem a compreensão total da aplicação antes de sua implantação para sua execução eficaz.
- Procure solucionar as **Problematizações** propostas pois irão exercitar sua capacidade de solução de problemas e fixar os conhecimentos adquiridos.

- Por fim, busque documentar seu progresso e compartilhar seu conhecimento. Salve cuidadosamente cada laboratório que fizer pois servirá de material para estudo. Procure compartilhar o que aprendeu. Ao ensinar você irá aprender mais ainda.

## Sobre o Autor

**Thiago Guimarães Tavares** é professor do Instituto Federal de Ciências e Tecnologia, *Campus Palmas* - TO. Mestre em Engenharia de Produção e Sistemas pela UNISINOS. Especialista em Redes de Computadores pela ESAB. Graduado em Redes de Computadores pelo CEULP/ULBRA.

Espero que este livro seja útil para seu processo de ensino ou aprendizagem. Você pode me encontrar em:

- Pessoal: [thiagogmta@gmail.com](mailto:thiagogmta@gmail.com)
- Profissional: [thiagogmta@ifto.edu.br](mailto:thiagogmta@ifto.edu.br)

# 1. Netkit - Sistema para Emulação de Redes de Computadores

Prof. Thiago Guimarães Tavares - [thiagogmta@gmail.com](mailto:thiagogmta@gmail.com) - IFTO Campus Palmas

### *Objetivos do capítulo*

Ao final deste capítulo você deverá ser capaz de:

- Entender o que é e para que serve o Netkit
- Aprender a realizar a instalação da ferramenta
- Criar uma rede ponto a ponto e um laboratório com 4 hosts e um hub

## 1.1. Apresentação

### 1.1.1. O que é?

Netkit é um sistema para emulação de redes de computadores que possibilita a criação de instâncias virtuais de vários dispositivos de uma rede. Cada dispositivo virtual possui:

- Um console (janela do xterminal).
- 32mb de consumo de memória ram. Esse valor pode ser alterado, mas é setado por padrão.
- Um sistema de arquivos armazenado em um único arquivo no disco com extensão .disk
- Uma ou várias interfaces de rede. Cada interface de rede pode ser conectada a um *domínio de emulação* (hub/switch).

### 1.1.2. Para que serve?

Redes de computadores são ambientes tipicamente complexos e são compostos por vários hosts, serviços, interfaces e protocolos em funcionamento. Por conta da diversidade de equipamentos e serviços o estudo prático da disciplina se torna dificultoso. Recriar um ambiente de redes real envolve alto custo. Existem alternativas para a simulação desses ambientes que possibilitam o estudo de várias ferramentas e protocolos. O Netkit é uma dessas ferramentas e serve justamente para criarmos um ambiente de redes de teste que necessita, para seu funcionamento, apenas de um computador com sistema operacional linux.

### 1.1.3. Como funciona?

A idéia básica do Netkit é que se possa criar várias máquinas virtuais de baixo consumo que possam ser conectadas a 'domínios de colisão virtuais' ou seja ligadas a hubs ou switches. Podemos manipular cada host através de um terminal do xterminal onde teremos permissão de administrador para realização de qualquer configuração, inclusive a configuração de servidores, roteadores, firewall etc.

## 1.2. Exemplificação



O Netkit funciona em ambientes linux. Para os nossos teste foi utilizado a distribuição Ubuntu.

### 1.2.1. Instalação

Inicialmente deve-se realizar o download dos três arquivos necessários para o funcionamento do Netkit. sendo eles:

- Netkit - Pacote com os aruivos para funcionamento do Netkit
- Netkit-filesystem - Pacote que contem uma imagem minima do sistema de arquivos da distribuição Debian incluindo vários pacotes instalados.
- Netkit-kernel - Módulo do kernel que permitirá emular o kernel do host real em uma istância do xtermianl.

Os arquivos se encontram em: <http://www.netkit.org/> e para os testes deste material foi utilizado a ultima versão estável a 2.8.

#### Baixando os pacotes

```
$ cd ~
$ wget http://wiki.netkit.org/download/netkit/netkit-2.8.tar.bz2
$ wget http://wiki.netkit.org/download/netkit-filesystem/netkit-filesystem-i386-
F5.2.tar.bz2
$ wget http://wiki.netkit.org/download/netkit-kernel/netkit-kernel-i386-K2.8.tar.bz2
```

#### Descompactando

```
$ tar -xjSf netkit-2.8.tar.bz2
$ tar -xjSf netkit-filesystem-i386-F5.2.tar.bz2
$ tar -xjSf netkit-kernel-i386-K2.8.tar.bz2
```

## Exportando Variáveis de Ambiente

As variáveis de ambiente são responsáveis por armazenar os caminhos dos arquivos de configuração do netkit

```
export NETKIT_HOME=~/netkit
export PATH=$PATH:$NETKIT_HOME/bin
export MANPATH=:$NETKIT_HOME/man
```

## Salvando em um arquivo

Podemos salvar as instruções que exportam as variáveis de ambiente em um arquivo para não precisar executar esses três comandos todas as vezes que o computador for ligado.

```
$ echo export NETKIT_HOME=~/netkit >> ~/.bashrc
$ echo export PATH=$PATH:$NETKIT_HOME/bin >> ~/.bashrc
$ echo export MANPATH=:$NETKIT_HOME/man >> ~/.bashrc
```

## Verificando Tudo

```
$ cd ~/netkit
$ ./check_configuration.sh
```

Ao verificar a configuração realizada no seu terminal deverá apresentar as seguintes informações

```
thiago@thiago: ~/netkit
thiago@thiago:~$ cd ~/netkit/
thiago@thiago:~/netkit$ ./check_configuration.sh
> Checking path correctness... passed.
> Checking environment... passed.
> Checking for availability of man pages... passed.
> Checking for proper directories in the PATH... passed.
> Checking for availability of auxiliary tools:
    awk      : ok
    basename : ok
    date     : ok
    dirname  : ok
    find     : ok
    getopt   : ok
    grep     : ok
    head     : ok
    id       : ok
    kill     : ok
    ls       : ok
    lsof     : ok
    ps       : ok
    readlink : ok
    wc       : ok
    port-helper : ok
    tuncctl  : ok
    uml_mconsole : ok
    uml_switch : ok
passed.
> Checking for availability of terminal emulator applications:
    xterm      : found
    konsole    : not found
    gnome-terminal : found
passed.
> Checking filesystem type... passed.
> Checking whether 32-bit executables can run... passed.
[ READY ] Congratulations! Your Netkit setup is now complete!
          Enjoy Netkit!
```

Figura 1. Parabéns! Sua instalação do Netkit está completa!

Caso apresente algum erro, poderá ser necessário a instalação de algumas dependência.

### Instalação de Dependências

```
$ sudo apt-get update
$ sudo apt-get install ia32-libs
$ sudo apt-get install libc6-i386
```

## 1.3. Utilizando o Ambiente

A seguir será exibido como inicializar um único host ou um laboratório com vários dispositivos bem como a abordagem dos comandos necessários.

### 1.3.1. Comandos

O Netkit possui dois grupos de comandos. Esses comandos podem ser executados no seu terminal linux sendo comandos com prefixo **v** e os comandos com prefixo **l**.

- **v**<comando>
- **l**<comando>

Os comandos com prefixo **v** são utilizados para configuração e inicialização de um único host virtual. Já os comandos com prefixo **l** são utilizados para manipulação de laboratórios com vários hosts.

#### **V**<comandos>

**vstart** Inicia uma nova maquina virtual cada novo host virtual instanciado pelo mesmo usuário

deve possuir um nome diferente. Caso parâmetros adicionais não sejam especificados o host virtual será inicializado conforme parâmetros especificados nos arquivos de configuração do netkit. Exemplo de funcionamento:

```
$ vstart pc1 ①
```

① Inicia um host virtual com nome: pc1.

Pode ser observado que ao iniciar um host virtual será criado um arquivo pc1.disk e esse arquivo será o sistema de arquivos do seu host virtual. A medida que configurações sejam feitas, estas serão armazenadas dentro desse disco virtual. Caso o disco virtual seja excluído as configurações serão perdidas. O host virtual poderá ser finalizado e inicializado novamente posteriormente através do mesmo comando.

Alguns parâmetros adicionais:

- --mem=xx - Define quantidade de memória ram a ser consumida pelo host virtual onde xx deve ser substituído pelo valor em mb.
- --ethx=DOMAIN - Defina uma interface de rede para o host virtual onde o x deve ser substituído pelo número da interface. O DOMAIN simboliza o domínio de colisão de pacotes (hub/switch) e deve ser substituído pelo nome do equipamento caso pertinente.
- --exec=COMMAND - Especifica comando a ser executado assim que o host virtual for iniciado.

Para mais parâmetros verifique o manual do netkit em <http://www.netkit.org/>.

Exemplos de utilização:

```
$ vstart pc1 --eth0=A --eth1=B --mem=100 ①
```

① Inicia um host virtual com nome: pc1 e duas interfaces de rede (eth0 e eth1) cada interface está conectada a um domínio de colisão sendo respectivamente A e B. O host em questão irá consumir 100mb de memória ram.



Você poderá iniciar um novo host a partir de qualquer diretório desde que este diretório não contenha espaços em seu nome. Caso isso ocorra um aviso de erro poderá ser exibido em relação ao caminho das variáveis de ambiente.

**vlist** Lista máquinas virtuais em atividade exibindo informações a seu respeito.

**vconfig** Habilita novas interfaces de rede para máquinas virtuais que já estejam em funcionamento.

```
$ vconfig --eth2=A pc1 ①
```

① Adiciona nova interface de rede (eth2) conectada ao domínio de colisão A ao pc1

**vcrash** Finaliza um ou vários hosts virtuais. Os hosts virtuais também podem ser desligados através



do comando halt dentro de seu próprio terminal.

```
$ vcrash pc1 ①
```

① finaliza a execução do host pc1

### L<comandos>

A seguir serão apresentados os comandos do grupo l para que possamos trabalhar com laboratórios. Entretanto, os comandos apenas surtirão efeito após a criação da estrutura de diretórios e arquivos para o laboratório.

**lstart** Inicia todos os hosts do laboratório

**lcrash** Finaliza todos os hosts do laboratório

**lclean** Após a finalização do laboratório o comando lclean pode ser utilizado para limpar eventuais arquivos temporários.

**linfo** Apresenta informações a respeito do laboratório sem que este ainda tenha sido inicializado.

## 1.3.2. Criando uma Rede Ponto-a-Ponto

Nessa sessão criaremos uma rede ponto-a-ponto através de dois hosts virtuais conectados a um domínio de colisão. Execute os comandos a seguir no terminal:

```
$ vstart pc1 --eth0=A ①  
$ vstart pc2 --eth0=A ②
```

① Inicializa um host virtual (pc1) com uma interface de rede (eth0) conectada ao domínio A

② Inicializa um segundo host virtual (pc2) com uma interface de rede (eth0) conectada ao mesmo domínio A

Para melhor compreensão criamos dois hosts ligados a uma espécie de hub virtual. Nesse ambiente de testes os dois hosts estão fisicamente conectados. Para que possam trocar informação basta configurarmos os endereços em suas respectivas interfaces de rede. Podemos fazer isso através do comando **ifconfig**. Logo abaixo temos dois comandos que devem ser executados cada um em uma janela de terminal dos hosts virtuais.

```
pc1$ ifconfig eth0 192.168.10.1 ①  
pc2$ ifconfig eth0 192.168.10.2 ②
```

① Atribui o endereço 192.168.10.1 ao pc1

② Atribui o endereço 192.168.10.2 ao pc2

Para testarmos a comunicação podemos enviar um pacote de teste do pc1 para o pc2. Utilizaremos dois comandos um para enviar os pacotes através do pc1 o comando **ping** e outro para monitorar o

recebimento dos pacotes no pc2 o comando **tcpdump**.

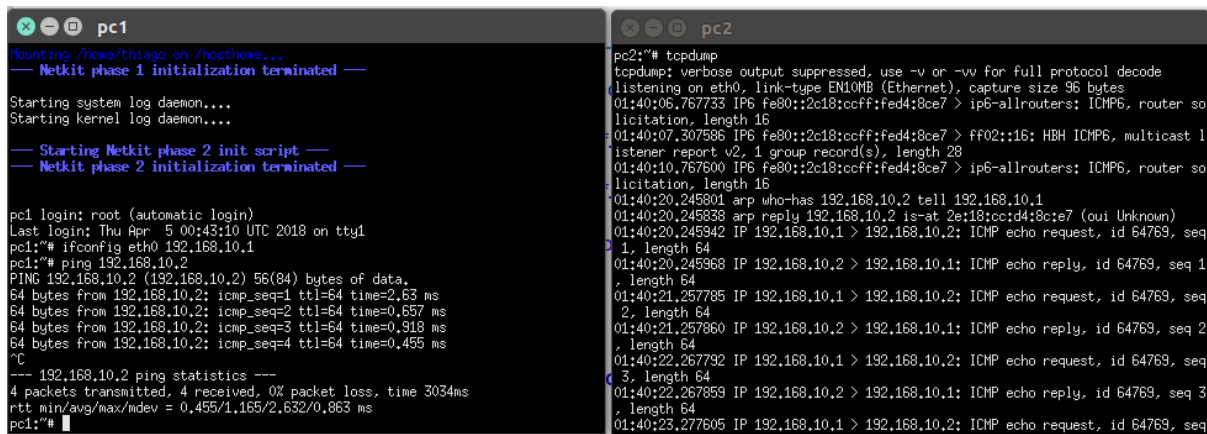
No terminal do pc2 execute

```
$ tcpdump
```

No terminal do pc1 execute

```
$ ping 192.168.10.2
```

A imagem a seguir ilustra o funcionamento do teste onde o pc1 envia pacotes ao pc2 enquanto este monitora o recebimento.



```
pc1
Mounting /home/thiago on /hosthome...
--- Netkit phase 1 initialization terminated ---
Starting system log daemon....
Starting kernel log daemon....
--- Starting Netkit phase 2 init script ---
--- Netkit phase 2 initialization terminated ---

pc1 login: root (automatic login)
Last login: Thu Apr  5 00:43:10 UTC 2018 on tty1
pc1:~# ifconfig eth0 192.168.10.1
pc1:~# ping 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data:
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=2.63 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=64 time=0.657 ms
64 bytes from 192.168.10.2: icmp_seq=3 ttl=64 time=0.318 ms
64 bytes from 192.168.10.2: icmp_seq=4 ttl=64 time=0.455 ms
^C
--- 192.168.10.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3034ms
rtt min/avg/max/mdev = 0.455/1.165/2.632/0.863 ms
pc1:~#

pc2
pc2:~# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
01:40:06.767733 IP6 fe80::2c18:ccff:fed4:8ce7 > ip6-allrouters: ICMP6, router so
licitation, length 16
01:40:07.307586 IP6 fe80::2c18:ccff:fed4:8ce7 > ff02::16: HBH ICMP6, multicast l
istener report v2, 1 group record(s), length 28
01:40:10.767600 IP6 fe80::2c18:ccff:fed4:8ce7 > ip6-allrouters: ICMP6, router so
licitation, length 16
01:40:20.245801 arp who-has 192.168.10.2 tell 192.168.10.1
01:40:20.245838 arp reply 192.168.10.2 is-at 2e:18:cc:d4:8c:e7 (oui Unknown)
01:40:20.245942 IP 192.168.10.1 > 192.168.10.2: ICMP echo request, id 64769, seq
1, length 64
01:40:20.245968 IP 192.168.10.2 > 192.168.10.1: ICMP echo reply, id 64769, seq 1
, length 64
01:40:21.257785 IP 192.168.10.1 > 192.168.10.2: ICMP echo request, id 64769, seq
2, length 64
01:40:21.257860 IP 192.168.10.2 > 192.168.10.1: ICMP echo reply, id 64769, seq 2
, length 64
01:40:22.267792 IP 192.168.10.1 > 192.168.10.2: ICMP echo request, id 64769, seq
3, length 64
01:40:22.267859 IP 192.168.10.2 > 192.168.10.1: ICMP echo reply, id 64769, seq 3
, length 64
01:40:23.277605 IP 192.168.10.1 > 192.168.10.2: ICMP echo request, id 64769, seq
```

Figura 2. Teste rede ponto-a-ponto

### 1.3.3. Criando um Laboratório

Um laboratório no netkit é um set de configurações para hosts virtuais que podem ser iniciados e finalizados simultaneamente. Para tanto deve ser criada uma estrutura de arquivos e diretórios. Sendo:

- Lab.conf - Arquivo de configuração para descrever a estrutura e topologia do laboratório
- Diretórios - Um diretório para cada host virtual que poderá armazenar qualquer arquivo a ser utilizado por aquele host específico. Esse diretório funcionará como o diretório raiz (/) para o host. Dessa forma, caso seja criado um arquivo ou outros diretórios ali dentro, estes poderão ser visualizados dentro do terminal do host virtual.
- .startup e .shutdown - Cada host virtual poderá ter o seu arquivo .startup ou .shutdown e esses arquivos servem para respectivamente armazenarem instruções que serão processadas assim que o host for iniciado ou desligado.

Para nosso exemplo utilizaremos como cenário 3 hosts e dois hubs. O pc1 e 3 terão apenas uma interface de rede e estarão conectados respectivamente ao hub 01 e 02. Já o pc2 terá duas interfaces de rede e será conectado aos dois hubs. Segue diagrama do nosso cenário.

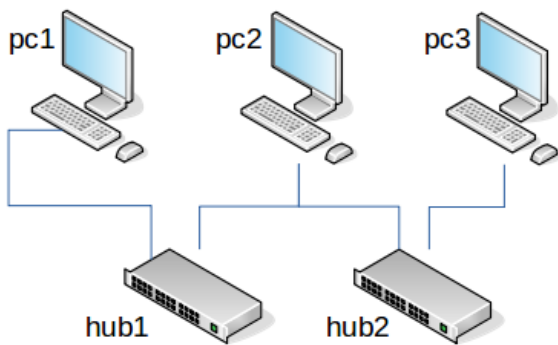


Figura 3. Laboratório 01

## 1 - Criando arquivos e diretórios

Crie um diretório para armazenar o laboratório. Dentro desse diretório serão criados um diretório para cada host, um arquivo .startup para cada host e um arquivo .conf para o laboratório.



O nome do diretório que irá armazenar o laboratório não deve conter espaços.

```
$ cd ~
$ mkdir lab01
$ cd lab01
$ touch lab.conf pc1.startup pc2.startup pc3.startup
$ mkdir pc1 pc2 pc3
```

## 2 - Realizando configurações

Apos criados os arquivos e diretórios iremos inserir as configurações.

```
#lab.conf
pc1[0]=hub1
pc2[0]=hub1
pc2[1]=hub2
pc3[0]=hub2

pc2[mem]=64
```

A lógica do lab.conf é a seguinte: O pc1 está conectado ao hub1 através da interface 0. O pc2 está conectado ao hub1 via interface 0 e ao hub2 via interface 1. Já o pc3 conecta-se ao hub2 via interface 0. Caso seja necessário alterar o tamanho de memória a ser consumido pelo host pode ser feito conforme a última linha do arquivo. A quantidade de memória padrão (32mb) já é suficiente para o funcionamento do host virtual.

```
#pc1.startup
ifconfig eth0 192.168.10.1
```

```
#pc2.startup
ifconfig eth0 192.168.10.2
ifconfig eth1 10.0.0.1
```

```
#pc3.startup
ifconfig eth0 10.0.0.2
```

Os arquivos .startup inserem configurações assim que o host for iniciado. Pode ser utilizado qualquer instrução linux para criarmos diversas configurações. Para os três arquivos apresentados foi utilizado o comando ifconfig para configurar as interfaces com os endereços escolhidos.

### 3 - Subindo o Laboratório

Dentro do diretório onde se encontram os arquivos do laboratório execute o seguinte comando:

```
$ lstart
```

Seu laboratório será iniciado e três janelas do xterminal serão abertas. Uma para cada host virtual. Realize alguns testes de comunicação entre os hosts. Em teoria o pc1 podera comunicar-se com o pc2 e este poderá enviar pacotes ao pc1 e ao pc3.

#### 1.3.4. Troca de arquivos entre host real e host virtual

Dentro de cada host virtual existe um diretório chamado /hosthome que pode ser acessado via interface xterminal. Esse diretório aponta diretamente para o diretório /home do host real. Dessa forma é possível acessar, através do host virtual, os arquivos que estão armazenados no diretório /home do host principal.



O diretório /hosthome é montado com permissão de leitura e escrita, tenha ciência das atividades a serem realizadas dentro deste diretório.

#### 1.3.5. Acesso a Internet através do Host virtual

É possível a criação de um túnel (tap interface) entre o host real e o host virtual de forma que o host virtual acesse redes externas através do host real.

##### Acesso a rede externa com vstart

```
vstart pc1 --eth0=tap,10.0.0.1,10.0.0.2 --eth1=A
```

O comando a cima inicia uma nova máquina virtual de nome pc1. Essa máquina irá ter duas interfaces de rede eth0 e eth1. A interface eth0 será configurada automaticamente com o endereço 10.0.0.2 quando a máquina for iniciada. O endereço ip 10.0.0.1 será utilizado para rotear o tráfego entre o host real e o host virtual. Para utilização do tap ambos os endereços devem estar na mesma faixa. Ao executar este comando a senha de root deve ser solicitada.

## Acesso a rede externa em um laboratório

Para configurar um acesso a rede externa dentro de um laboratório. Basta fazer acesso ao arquivo lab.conf e inserir a instrução correspondente ao tap ao pc desejado. Como mostra a instrução a baixo.

```
#lab.conf  
pc1[0]=tap,192.168.10.1,192.168.10.2
```

## 1.4. Criação do ambiente e Informações Iniciais

O Apache é um serviço que oferece suporte a hospedagem de um ou vários sites no servidor, interpretando páginas estáticas e scripts. Para que seja possível a hospedagem de páginas dinâmicas com integração com PHP e MySQL por exemplo, faz-se necessário a implementação do LAMP. O apache pode ser dividido em dois grupos de versões mais utilizadas sendo a versão 1.3 e a versão 2 qual será utilizada neste guia. O motivo da versão 1.3 ainda ser utilizada é sua compatibilidade com módulos antigos não disponíveis para versão 2.

### Criação do Ambiente

Leve em consideração que nosso ambiente tem a seguinte topologia implementada através do netkit:

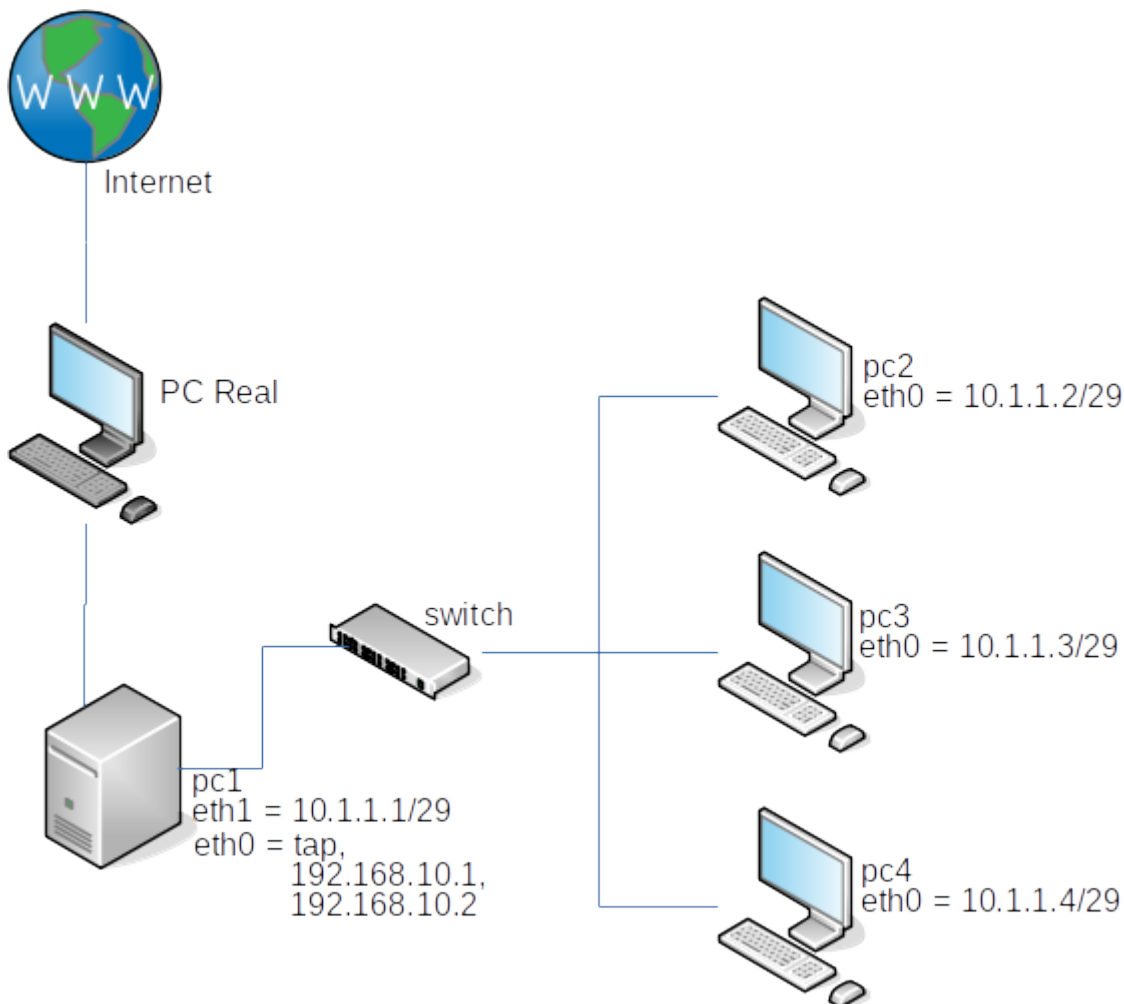


Figura 4. Laboratório 01

## 2. Dynamic Host Configuration Protocol (DHCP)

Prof. Thiago Guimarães Tavares - [thiagogmta@gmail.com](mailto:thiagogmta@gmail.com) IFTO Campus Palmas

### Objetivos do capítulo

Ao final deste capítulo você deverá ser capaz de:

- Entender o que é e para que serve um servidor DHCP.
- Ser capaz de realizar a configuração de um servidor DHCP simples.
- Realizar o vínculo de Endereços MACs a determinados endereços IP's através do servidor DHCP.

### 2.1. Apresentação: DHCP

#### 2.1.1. O que é?

A sigla **DHCP** representa Dynamic Host Configuration Protocol ou Protocolo de Configuração de

Host Dinâmico. O DHCP é responsável pela distribuição de endereços Ips (e outras informações) aos hosts da rede (MORIMOTO, 2013).

### **2.1.2. Para que serve?**

O DHCP automatiza a configuração da rede e, de acordo com sua configuração, otimiza a utilização de endereços Ips, destinando apenas à quantidade de hosts necessários. Quando um novo host é adicionado na rede este host solicita ao DHCP um endereço IP válido. O DHCP ao responder a solicitação entrega um endereço apropriado e outras informações como máscara, gateway e DNS (MORIMOTO, 2013).

### **2.1.3. Como funciona?**

A estação de trabalho faz uma requisição em broadcast perguntando se existe alguém que possa lhe atribuir um endereço válido. O servidor DHCP por sua vez responde às solicitações vindas pela rede (atribuídas a ele) fornecendo um IP válido e demais configurações necessárias. A resposta do servidor DHCP é feita via broadcast endereçada ao IP 255.255.255.255 e é retransmitida pelo switch a todos os hosts (MORIMOTO, 2013).

De tempos em tempos o servidor DHCP verifica se os endereços IPs atribuídos ainda estão em uso, evitando o desperdício de endereços (MORIMOTO, 2013).

## **2.2. DHCP: Exemplificação**

### **2.2.1. Criação do Ambiente**

Leve em consideração que nosso ambiente tem a seguinte topologia implementada através do netkit:

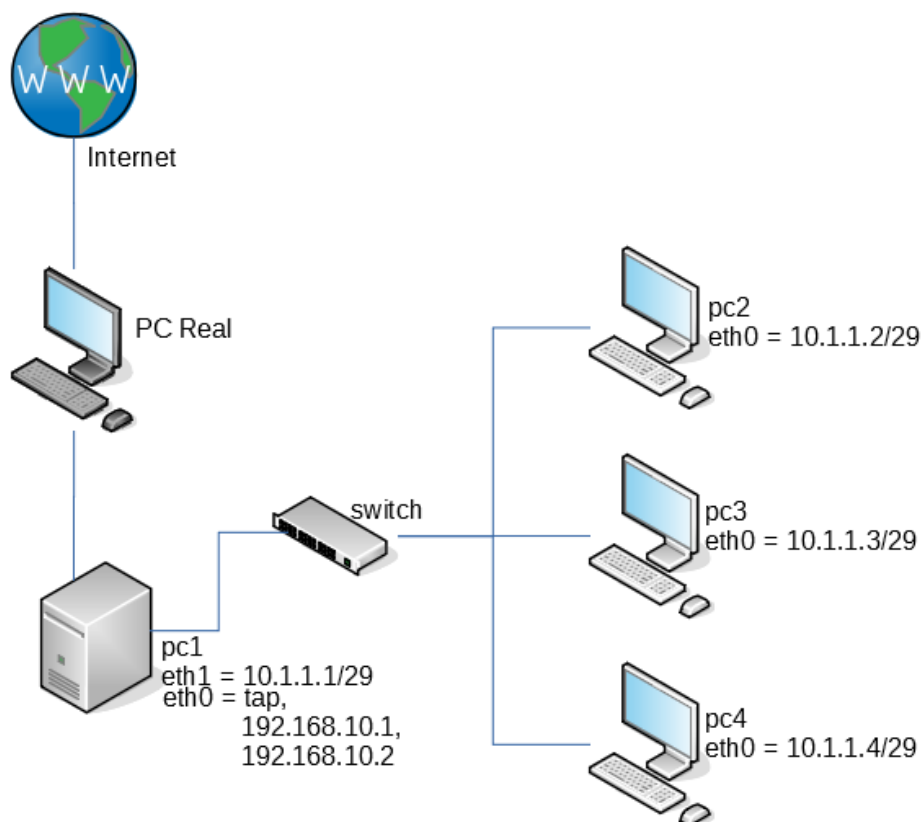


Figura 5. Laboratório 01

### Etapa 1: Criando Arquivos e Diretórios

Primeiro passo iremos criar os arquivos e diretórios necessários. Crie um diretório chamado **lab** para que você possa guardar os seus laboratórios. Os procedimentos logo a baixo partem de um diretório lab previamente criado.



```
$ mkdir dhcp
$ cd dhcp
$ mkdir pc1 pc2 pc3 pc4
$ touch lab.conf pc1.startup pc2.startup pc3.startup pc4.startup
```

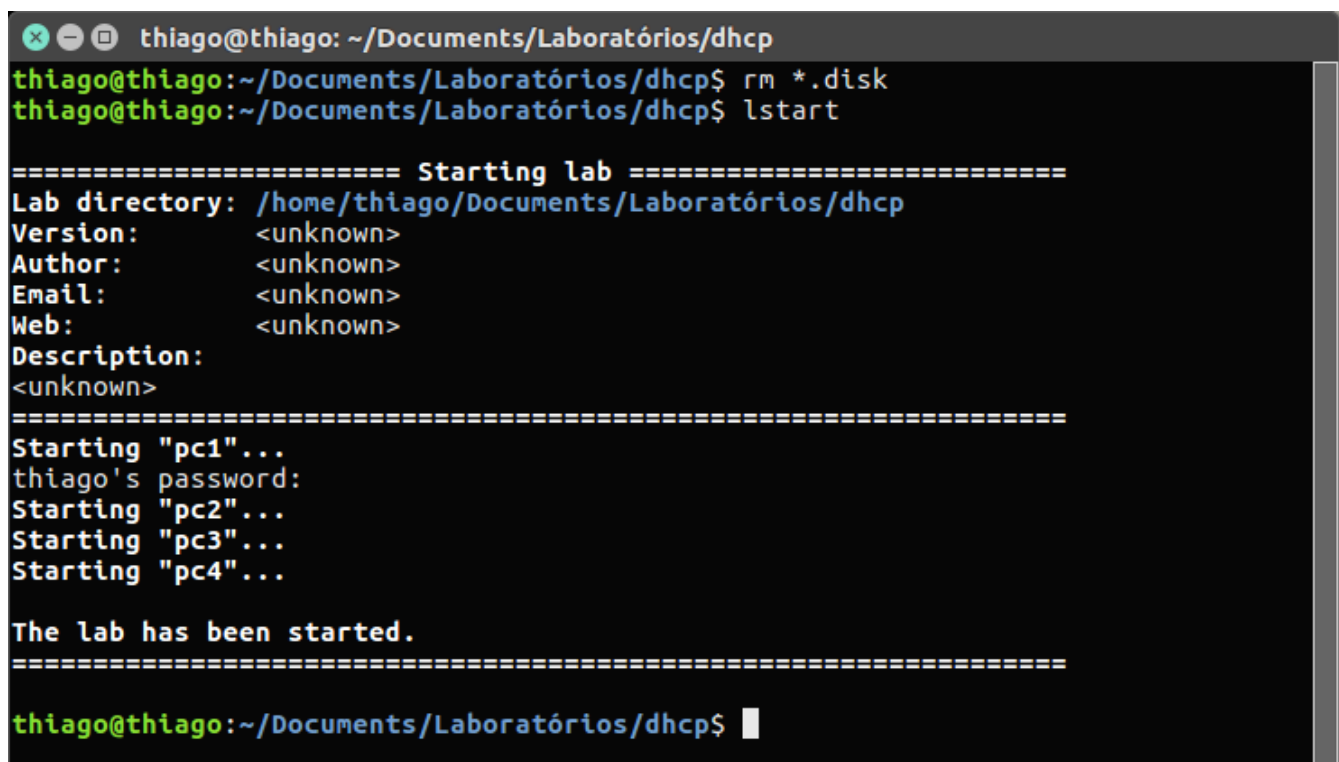
## Etapa 2: Configurando Topologia

Iremos configurar a topologia do laboratório através do arquivo lab.conf. Para tanto insira o seguinte conteúdo no arquivo:

```
#lab.conf
pc1[0]=tap,192.168.10.1,192.168.10.2
pc1[1]=switch
pc2[0]=switch
pc3[0]=switch
pc4[0]=switch
```

Configurações feitas, inicie o laboratório com o comando:

```
$ lstart
```



```
thiago@thiago: ~/Documents/Laboratórios/dhcp
thiago@thiago:~/Documents/Laboratórios/dhcp$ rm *.disk
thiago@thiago:~/Documents/Laboratórios/dhcp$ lstart

===== Starting lab =====
Lab directory: /home/thiago/Documents/Laboratórios/dhcp
Version:      <unknown>
Author:       <unknown>
Email:        <unknown>
Web:          <unknown>
Description:  <unknown>
=====
Starting "pc1"...
thiago's password:
Starting "pc2"...
Starting "pc3"...
Starting "pc4"...

The lab has been started.
=====

thiago@thiago:~/Documents/Laboratórios/dhcp$
```

Figura 6. Iniciando o Laboratório

Os quatro hosts virtuais serão inicializados.

```
pc1
Apr 18 01:48:18 Exception: Resolver binding to server socket on port 53 for 127.0.0.1: Address already in use
[....] Starting ftp server: proftpdpc1 proftpd[2259]: mod_tls/2.4.3: compiled using OpenSSL version 'OpenSSL 1.0.1e 11 Feb 2013' headers, but linked to OpenSSL version 'OpenSSL 1.0.1t 3 May 2016' library
pc1 proftpd[2259]: mod_sftp/0.9.8: compiled using OpenSSL version 'OpenSSL 1.0.1e 11 Feb 2013' headers, but linked to OpenSSL version 'OpenSSL 1.0.1t 3 May 2016' library
pc1 proftpd[2259]: mod_tls_memcache/0.1: notice: unable to register 'memcache' SSL session cache; Memcache support not enabled
* ok
Loading capability module if not yet done.
Starting Quagga daemons (prio:10):.
[ ok ] Starting Squid HTTP proxy: squid.
[ ok ] Starting OpenBSD Secure Shell server: sshd.
[FAIL] startpar: service(s) returned failure: dnsmasq isc-dhcp-server ... failed

pc1 login: root (automatic login)
Last login: Thu Jan 25 13:00:44 CET 2018 on tty0
Linux update 3.2.54-netkit-ng-K3.2 #2 Tue Nov 11 11:42:04 CET 2014 i686
Welcome to Netkit

root@pc1:~#

pc4
Apr 18 01:48:51 Exception: Resolver binding to server socket on port 53 for 127.0.0.1: Address already in use
[....] Starting ftp server: proftpdpc4 proftpd[2247]: mod_tls/2.4.3: compiled using OpenSSL version 'OpenSSL 1.0.1e 11 Feb 2013' headers, but linked to OpenSSL version 'OpenSSL 1.0.1t 3 May 2016' library
pc4 proftpd[2247]: mod_sftp/0.9.8: compiled using OpenSSL version 'OpenSSL 1.0.1e 11 Feb 2013' headers, but linked to OpenSSL version 'OpenSSL 1.0.1t 3 May 2016' library
pc4 proftpd[2247]: mod_tls_memcache/0.1: notice: unable to register 'memcache' SSL session cache; Memcache support not enabled
* ok
Loading capability module if not yet done.
Starting Quagga daemons (prio:10):.
[ ok ] Starting Squid HTTP proxy: squid.
[ ok ] Starting OpenBSD Secure Shell server: sshd.
[FAIL] startpar: service(s) returned failure: dnsmasq isc-dhcp-server ... failed

pc4 login: root (automatic login)
Last login: Thu Jan 25 13:00:44 CET 2018 on tty0
Linux update 3.2.54-netkit-ng-K3.2 #2 Tue Nov 11 11:42:04 CET 2014 i686
Welcome to Netkit

root@pc4:~#

pc2
Apr 18 01:48:30 Exception: Resolver binding to server socket on port 53 for 127.0.0.1: Address already in use
[....] Starting ftp server: proftpdpc2 proftpd[2234]: mod_tls/2.4.3: compiled using OpenSSL version 'OpenSSL 1.0.1e 11 Feb 2013' headers, but linked to OpenSSL version 'OpenSSL 1.0.1t 3 May 2016' library
pc2 proftpd[2234]: mod_sftp/0.9.8: compiled using OpenSSL version 'OpenSSL 1.0.1e 11 Feb 2013' headers, but linked to OpenSSL version 'OpenSSL 1.0.1t 3 May 2016' library
pc2 proftpd[2234]: mod_tls_memcache/0.1: notice: unable to register 'memcache' SSL session cache; Memcache support not enabled
* ok
Loading capability module if not yet done.
Starting Quagga daemons (prio:10):.
[ ok ] Starting Squid HTTP proxy: squid.
[ ok ] Starting OpenBSD Secure Shell server: sshd.
[FAIL] startpar: service(s) returned failure: dnsmasq isc-dhcp-server ... failed

pc2 login: root (automatic login)
Last login: Thu Jan 25 13:00:44 CET 2018 on tty0
Linux update 3.2.54-netkit-ng-K3.2 #2 Tue Nov 11 11:42:04 CET 2014 i686
Welcome to Netkit

root@pc2:~#

pc3
Apr 18 01:48:39 Exception: Resolver binding to server socket on port 53 for 127.0.0.1: Address already in use
[....] Starting ftp server: proftpdpc3 proftpd[2239]: mod_tls/2.4.3: compiled using OpenSSL version 'OpenSSL 1.0.1e 11 Feb 2013' headers, but linked to OpenSSL version 'OpenSSL 1.0.1t 3 May 2016' library
pc3 proftpd[2239]: mod_sftp/0.9.8: compiled using OpenSSL version 'OpenSSL 1.0.1e 11 Feb 2013' headers, but linked to OpenSSL version 'OpenSSL 1.0.1t 3 May 2016' library
pc3 proftpd[2239]: mod_tls_memcache/0.1: notice: unable to register 'memcache' SSL session cache; Memcache support not enabled
* ok
Loading capability module if not yet done.
Starting Quagga daemons (prio:10):.
[ ok ] Starting Squid HTTP proxy: squid.
[ ok ] Starting OpenBSD Secure Shell server: sshd.
[FAIL] startpar: service(s) returned failure: dnsmasq isc-dhcp-server ... failed

pc3 login: root (automatic login)
Last login: Thu Jan 25 13:00:44 CET 2018 on tty0
Linux update 3.2.54-netkit-ng-K3.2 #2 Tue Nov 11 11:42:04 CET 2014 i686
Welcome to Netkit

root@pc3:~#
```

Figura 7. Laboratório DHCP Iniciado

## 2.2.2. Configuração do Servidor DHCP

Nas distribuições linux baseadas em debian podemos instalar o pacote *isc-dhcp-server* correspondente ao DHCP via apt-get. Para a instalação do serviço utilize o seguinte comando:

```
$ apt-get install isc-dhcp-server
```

**Obs.:** Por padrão o *isc-dhcp* já vem instalado no netkit

Para iniciar, parar ou reiniciar o serviço utilize respectivamente:

```
$ /etc/init.d/isc-dhcp-server start
$ /etc/init.d/isc-dhcp-server stop
$ /etc/init.d/isc-dhcp-server restart
```

O arquivo de configuração do servidor DHCP encontra-se em:

```
/etc/dhcp/dhcpd.conf
```

## Etapa 1: Configurando interface de rede do Servidor

Nosso servidor DHCP necessita de um IP fixo para funcionamento. Tomaremos como parâmetro o endereço ip 10.1.1.1 para nosso servidor conforme nosso diagrama de rede. No terminal do host pc1 (nosso servidor) execute:

```
$ ifconfig eth1 10.1.1.1/29
```

## Etapa 2: Editando o arquivo de configuração do DHCP

O arquivo de configuração padrão do dhcp contem várias instruções e comentários. Iremos fazer o backup do arquivo original e criar um novo arquivo apenas com as configurações necessárias. Execute os comandos a seguir no servidor dhcp.

```
#pc1
$ cd /etc/dhcp
$ cp dhcpd.conf dhcpd.conf.bk
$ echo > dhcpd.conf
```

Insira o seguinte conteúdo no arquivo dhcpd.conf

```
#/etc/dhcp/dhcpd.conf
ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;
authoritative;
subnet 10.1.1.0 netmask 255.255.255.248 {
    range 10.1.1.3 10.1.1.6;
    option subnet-mask 255.255.255.248;
    option routers 10.1.1.1;
    option domain-name-servers 8.8.8.8;
    option broadcast-address 10.1.1.7;
}
```

### Detalhamento da Configuração

Default-lease-time 600	Informa que o servidor dhcp irá verificar a cada 600 segundos (10 minutos) se as estações estão ativas.
Max-lease-time 7200	Determina o tempo máximo que um host pode ficar com um endereço
Subnet	Informa o endereço da rede e sua máscara e dentro de seu bloco são inseridas as configurações para distribuição de endereços.
Range	Determina a faixa de endereços IPs que serão distribuídos pelo servidor

Option subnet-mask	Define a mascara padrão da rede
Option routers	Define o endereço do gateway padrão
Option domain-name-server	Contem os endereços dos servidores DNS. Ao utilizar dois endereços deve-se separá-los por vírgula.
Option broadcast-address	O endereço de broadcast da rede

### > Interfaces

Como nosso servidor possui duas interfaces de rede precisamos informar ao servidor qual das duas interfaces irá responder às solicitações de endereçamento IP. Para isso devemos editar o arquivo */etc/default/isc-dhcp-server*. Dentro dessa linha alterar o parâmetro INTERFACES e informar a interface desejada ficando dessa maneira:

```
#/etc/default/isc-dhcp-server
INTERFACES="eth1"
```

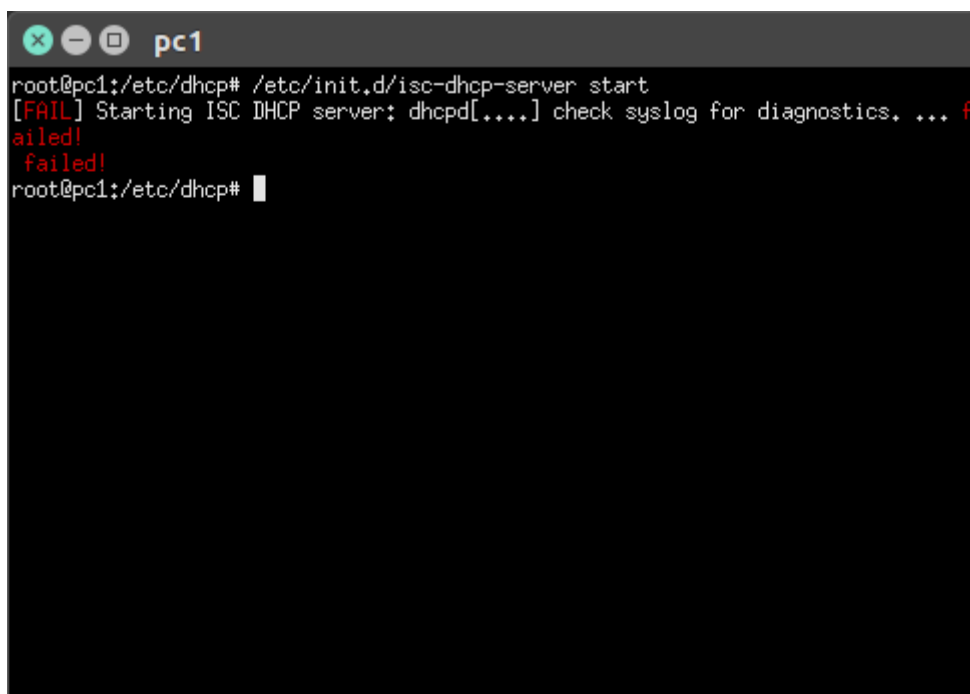
Salve e feche o arquivo.

### Etapa 3: Iniciando o Servidor

Configurações realizadas é hora de iniciar o servidor dhcp. Execute o comando:

```
$ /etc/init.d/isc-dhcp-server start
```

Caso ocorra algum erro de configuração será apresentado o erro e sua linha no arquivo de configuração nesse etapa. Caso o erro não seja decorrente do arquivo de configuração pode ser apresentada a seguinte janela:

A terminal window titled 'pc1' with standard window controls. The prompt is 'root@pc1:/etc/dhcp#'. The command entered is '/etc/init.d/isc-dhcp-server start'. The output shows a failure: '[FAIL] Starting ISC DHCP server: dhcpd[....] check syslog for diagnostics. ... failed! failed!'. The prompt returns to 'root@pc1:/etc/dhcp#'.

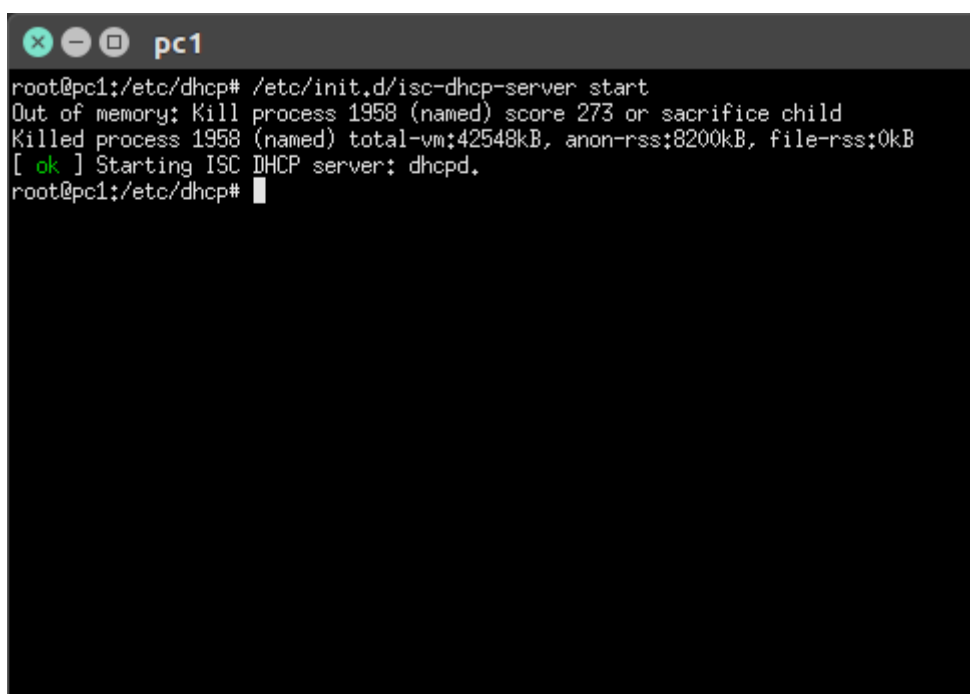
```
root@pc1:/etc/dhcp# /etc/init.d/isc-dhcp-server start
[FAIL] Starting ISC DHCP server: dhcpd[....] check syslog for diagnostics. ... failed!
failed!
root@pc1:/etc/dhcp#
```

Figura 8. Laboratório DHCP Iniciado

Para esses casos deve-se verificar o arquivo syslog.

```
$ cat /var/log/syslog
```

Entretanto, em teoria, seu laboratório deve funcionar perfeitamente apresentando a seguinte janela:

A terminal window titled 'pc1' with standard window controls. The prompt is 'root@pc1:/etc/dhcp#'. The command entered is '/etc/init.d/isc-dhcp-server start'. The output shows a successful start after a memory warning: 'Out of memory: Kill process 1958 (named) score 273 or sacrifice child Killed process 1958 (named) total-vm:42548kB, anon-rss:8200kB, file-rss:0kB [ ok ] Starting ISC DHCP server: dhcpd.'. The prompt returns to 'root@pc1:/etc/dhcp#'.

```
root@pc1:/etc/dhcp# /etc/init.d/isc-dhcp-server start
Out of memory: Kill process 1958 (named) score 273 or sacrifice child
Killed process 1958 (named) total-vm:42548kB, anon-rss:8200kB, file-rss:0kB
[ ok ] Starting ISC DHCP server: dhcpd.
root@pc1:/etc/dhcp#
```

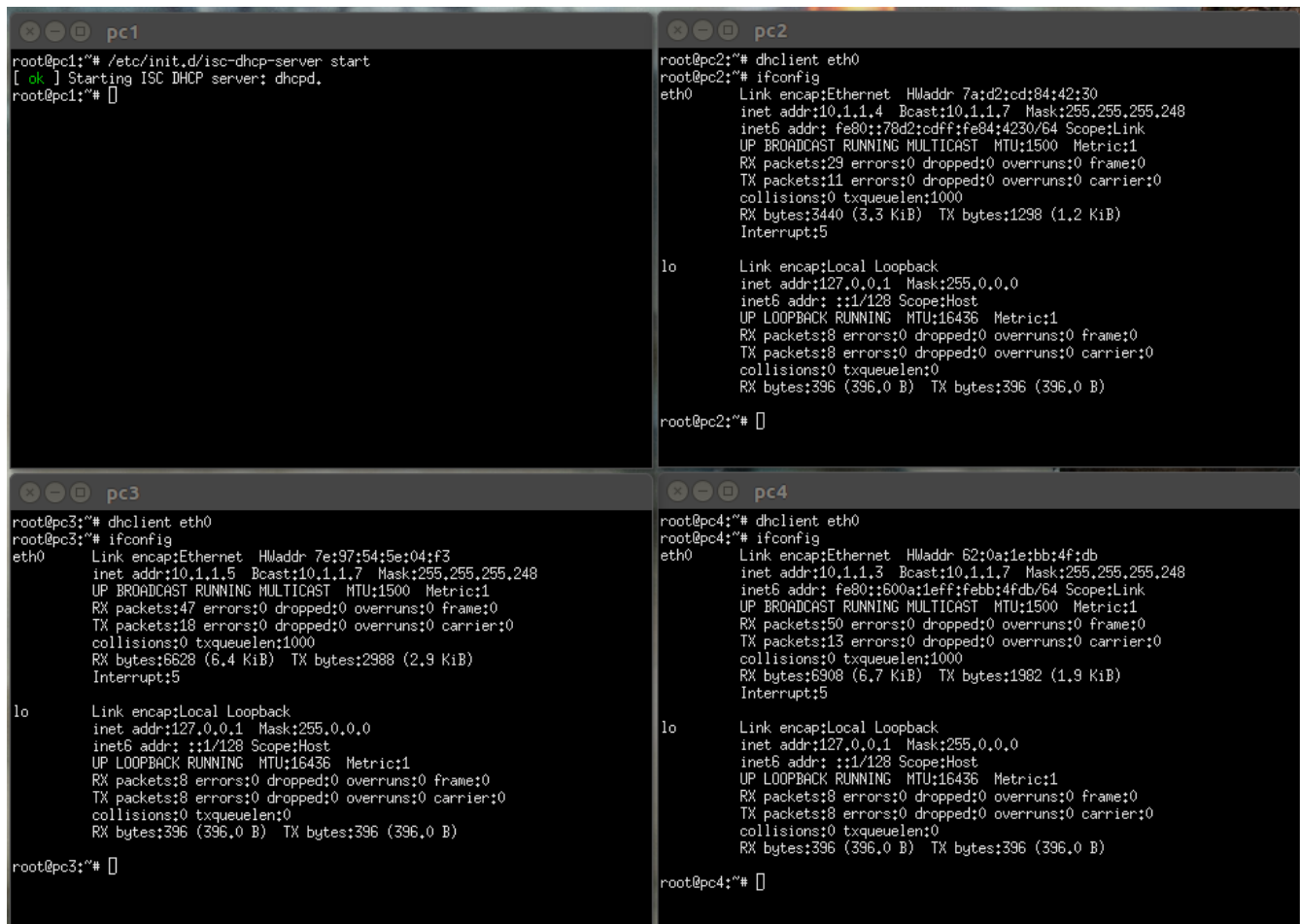
Figura 9. ISC DHCP OK

#### Etapa 4: Hosts Clientes

Com o servidor DHCP configurado os clientes já podem solicitar seus endereços IP. Para tanto

execute o seguinte comando nos hosts clientes (pc2, pc3 e pc4):

```
dhclient eth0
```



The image displays four terminal windows arranged in a 2x2 grid, each representing a different virtual host (pc1, pc2, pc3, pc4). Each window shows the execution of the `dhclient eth0` command and the subsequent output of the `ifconfig` command, which displays the network configuration for the `eth0` and `lo` interfaces.

**pc1:** Shows the DHCP server being started. The `ifconfig` output for `eth0` shows an IP address of `10.1.1.5` and a broadcast address of `10.1.1.7`. The `lo` interface shows a loopback address of `127.0.0.1`.

**pc2:** Shows the DHCP client configuration. The `ifconfig` output for `eth0` shows an IP address of `10.1.1.4` and a broadcast address of `10.1.1.7`. The `lo` interface shows a loopback address of `127.0.0.1`.

**pc3:** Shows the DHCP client configuration. The `ifconfig` output for `eth0` shows an IP address of `10.1.1.5` and a broadcast address of `10.1.1.7`. The `lo` interface shows a loopback address of `127.0.0.1`.

**pc4:** Shows the DHCP client configuration. The `ifconfig` output for `eth0` shows an IP address of `10.1.1.3` and a broadcast address of `10.1.1.7`. The `lo` interface shows a loopback address of `127.0.0.1`.

Figura 10. Distribuição de Endereços

Seu laboratório está com o servidor DHCP está pronto. Para validar as configurações realize testes de ping entre os hosts.

### 2.2.3. Fixando configurações no laboratório

Existe um pequeno impasse em relação ao funcionamento do laboratório. Quando o lab for finalizado com `lcrash` as configurações serão perdidas. Ao iniciar o laboratório novamente com `lstart` o laboratório será iniciado sem as configurações. Podemos realizar configurações nos diretórios dos hosts e nos arquivos `.startup` de forma que quando o laboratório seja iniciado as configurações já estejam presentes. Finalize seu laboratório atual e vamos aos procedimentos.

#### Etapa 1: Criando arquivos de configuração

Cada diretório de cada host virtual (pc1, pc2, pc3 e pc4) simboliza o ponto de montagem / dentro do terminal virtual daquele host. Ou seja, é possível criar uma estrutura de diretórios dentro do diretório do pc1 por exemplo e essa estrutura será apresentada no terminal do host pc1 partindo do diretório `/`.

Para o funcionamento do servidor DHCP é necessário a edição de seu arquivo de configuração e do

arquivo que indica a interface de rede que irá responder às requisições. Criaremos dois arquivos de configuração dentro do diretório do pc1, um referente ao servidor DHCP e o outro referente a interface de rede para que ambos apareçam dentro do host virtual. Os procedimentos a seguir devem ser executados no terminal do **host real**. Leve em consideração que deve-se estar dentro do diretório onde se encontram os arquivos do laboratório (lab/dhcp):

```
$ cd pc1
$ mkdir -p etc/dhcp
$ touch etc/dhcp/dhcpd.conf
```

Dentro do arquivo dhcpd.conf que foi criado insira o conteúdo do arquivo de configuração do servidor DHCP:

```
#etc/dhcp/dhcpd.conf
ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;
authoritative;
subnet 10.1.1.0 netmask 255.255.255.248 {
    range 10.1.1.3 10.1.1.6;
    option subnet-mask 255.255.255.248;
    option routers 10.1.1.1;
    option domain-name-servers 8.8.8.8;
    option broadcast-address 10.1.1.7;
}
```

Ainda dentro do diretório do pc1 iremos criar o arquivo responsável por informar a interface de rede que deve responder às requisições DHCP.

```
$ mkdir etc/default
$ touch etc/default/isc-dhcp-server
```

Dentro do arquivo isc-dhcp-server insira a seguinte instrução:

```
#etc/default/isc-dhcp-server
INTERFACES="eth1"
```

Ao iniciar o laboratório esses dois arquivos criados tomarão o lugar dos arquivos originais. Caso seja necessário alterar alguma configuração no arquivo de configuração do servidor, realize essa configuração no arquivo que foi criado nesta sessão para que quando o laboratório for inicializado as instruções já estejam presentes.

## Etapa 2: Configurando hosts

O servidor DHCP deve iniciar o serviço e os hosts clientes devem solicitar um endereço ao servidor assim que o lab for iniciado. Podemos inserir comandos nos arquivos .startup para que essas ações

sejam executadas. Seguem as configurações de cada arquivo:

```
#pc1.startup
ifconfig eth1 10.1.1.1/29
/etc/init.d/isc-dhcp-server start
```

```
#pc2.startup
dhclient eth0
```

```
#pc3.startup
dhclient eth0
```

```
#pc4.startup
dhclient eth0
```

Pronto basta iniciar o laboratório e todas as configurações já estarão presentes. Os quatro hosts do seu laboratório devem iniciar, o servidor já deve startar o serviço DHCP e os três clientes devem receber endereços IP's do servidor.

## 2.3. Problematização

1. Configure o seu servidor DHCP de forma a atribuir os seguintes endereços:

```
Faixa de endereço IP de 10.1.1.10 a 10.1.1.62
Máscara /26
Gateway: Endereço IP do servidor DHCP
DNS: 8.8.8.8, 8.8.4.4
```

2. Adicione mais um host : pc5
3. Encerre o laboratório com lcrash e certifique-se de que todos os arquivos .disk foram excluídos.
4. Envie na plataforma seu laboratório (não compacte)!
  - **Obs1:** Para fins de avaliação o laboratório deve estar totalmente funcional ao ser iniciado com lstart no computador do professor.
  - **Obs2:** Laboratórios que estiverem com os arquivos .disk ou que estiverem compactados, não serão corrigidos!

Envie na plataforma

## 3. Servidores DHCP e DNS

Prof. Thiago Guimarães Tavares - [thiagogmta@gmail.com](mailto:thiagogmta@gmail.com) IFTO Campus Palmas



Ao final deste capítulo você deverá ser capaz de:

- Entender o que é e para que serve um servidor DHCP.
- Ser capaz de realizar a configuração de um servidor DHCP simples.
- Realizar o vínculo de Endereços MACs a determinados endereços IP's através do servidor DHCP.

### 3.1. Apresentação deste material

Este guia apresenta informações sobre como realizar a configuração básica de dois serviços muito utilizado em redes de computadores. São os protocolos DHCP e DNS. O protocolo DHCP atua recebendo solicitações de endereços Ips advindos de outros hosts na rede. Já o servidor DNS atua realizando a conversão de nomes em endereços Ip's e vice-versa. Esses dois serviços (DHCP e DNS) podem ser configurados separadamente ou podem atuar em conjunto através do serviço DDNS que também será abordado. O guia está organizado em dois capítulos cada um abordando um serviço com exemplos práticos de configuração e trechos de arquivos de configuração.

## 4. Dynamic Host Configuration Protocol (DHCP)

### O que é?

A sigla **DHCP** representa Dynamic Host Configuration Protocol ou Protocolo de Configuração de Host Dinâmico. O DHCP é responsável pela distribuição de endereços Ips (e outras informações) aos hosts da rede (MORIMOTO, 2013).

### Para que serve?

O DHCP automatiza a configuração da rede e, de acordo com sua configuração, otimiza a utilização de endereços Ips, destinando apenas à quantidade de hosts necessários. Quando um novo host é adicionado na rede, este solicita ao DHCP um endereço IP válido. O DHCP ao responder a solicitação entrega um endereço apropriado e outras informações como máscara e gateway e DNS (MORIMOTO, 2013).

### Como funciona?

A estação de trabalho faz uma requisição em broadcast perguntando se existe alguém que possa lhe atribuir um endereço válido. O servidor DHCP por sua vez responde às solicitações vindas pela rede (atribuídas a ele) fornecendo um IP válido e demais configurações necessárias. A resposta do servidor DHCP é feita via broadcast endereçada ao IP 255.255.255.255 e é retransmitida pelo switch a todos os hosts (MORIMOTO, 2013). De tempos em tempos o servidor DHCP verifica se os endereços IPs atribuídos ainda estão em uso, evitando o desperdício de endereços (MORIMOTO, 2013).

## 4.1. DHCP: Instalação e Configuração

### 4.1.1. Instalando, comandos e arquivos

Nas distribuições linux baseadas em debian podemos instalar o pacote *dhcp3-server* correspondente ao DHCP via apt-get. Para a instalação do serviço utilize o seguinte comando:

```
$ apt-get install dhcp3-server
```

Para iniciar, parar ou reiniciar o serviço utilize respectivamente:

```
$ /etc/init.d/dhcp3-server start  
$ /etc/init.d/dhcp3-server stop  
$ /etc/init.d/dhcp3-server restart
```

O arquivo de configuração do servidor DHCP encontra-se em:

```
/etc/dhcp3/dhcpd.conf
```

### 4.1.2. Configurações Iniciais

## 4.2. DHCP: Destinando Ip fixo à um host

Teste

## 4.3. DHCP: Problematização



Para o desenvolvimento deste material foi utilizado como base o linux Ubuntu (seguem os mesmos procedimentos para Debian). Dessa maneira foi necessário a instalação do pacote *bind9*. Caso esteja utilizando outra distribuição busque como realizar a instalação do pacote *bind* (ou equivalente) em sua distribuição.

## 5. Domain Name System (DNS)

O que é?

**DNS** é a sigla para *Domain Name Server* ou Servidor de Nome de Domínio e pode ser comparado a uma base de dados. Esses servidores tem a função de receber requisições para resolução de nomes de domínios e retornar com os endereços IP's. Ao digitar em seu navegador de Internet o endereço *www.google.com.br* é feita uma requisição ao servidor DNS em busca do endereço do servidor que hospeda o site citado. O servidor DNS por sua vez verifica em sua base de dados se aquele endereço é conhecido. Em caso positivo o servidor retorna ao cliente a resposta de sua requisição. Em caso negativo o servidor encaminha essa solicitação à um servidor DNS superior.

## Para que serve?

Quando você visita um site através do seu navegador e digita seu endereço como por exemplo `www.google.com.br` é feita uma solicitação de acesso aos servidores que hospedam aquele website. Entretanto, o nome do website é uma forma de representação do endereço correspondente a sua hospedagem. A informação da localização destes servidores está em um servidor chamado DNS. Ou seja, a cada solicitação de acesso feita pelo navegador de internet é necessário a conversão do nome `www.google.com.br` para o endereço IP do servidor que o hospeda.

## Como funciona?

Quando você visita um site através do seu navegador e digita seu endereço como por exemplo `www.google.com.br` é feita uma solicitação de acesso aos servidores que hospedam aquele website. Entretanto, o nome do website é uma forma de representação do endereço correspondente a sua hospedagem. A informação da localização destes servidores está em um servidor chamado DNS. Ou seja, a cada solicitação de acesso feita pelo navegador de internet é necessário a conversão do nome `www.google.com.br` para o endereço IP do servidor que o hospeda.

## 5.1. DNS: Instalação e configuração

Teste

## 5.2. DNS: Servidor DNS secundário

Teste

## 5.3. DNS: Servidor DNS Dinâmico

Teste

## 5.4. DNS: Problematização DDNS

Teste

```
# sudo apt-get install bind9
```

① Instalação do bacote

# 6. Configurando servidor Dynamic DNS

Prof. Thiago Guimarães Tavares - [thiagogmta@gmail.com](mailto:thiagogmta@gmail.com) IFTO Campus Palmas

Ao final deste capítulo você deverá ser capaz de:

- Compreender os detalhes de funcionamento do servidor DDNS e sua relação com o DHCP e DNS
- Realizar a configuração do servidor DDNS em ambiente Linux
- Realizar testes para atualização automática das tabelas de nomes do servidor DNS

## 6.1. Apresentação: DDNS

### 6.1.1. O que é?

DDNS é abreviação para Dynamic DNS, ou Sistema de Nomes de Domínios Dinâmico. É um método utilizado para atualizar automaticamente um servidor de nomes de domínio.

### 6.1.2. Para que serve?

O DDNS é um complemento ao DNS e tem como objetivo relacionar um nome de domínio a um determinado usuário/host e manter esse relacionamento mesmo que o endereço IP desse usuário/host seja alterado. Podemos apresentar o exemplo de uma conexão de internet comum em uma residência. O modem do usuário irá receber um endereço de IP fixo do provedor para que possa acessar a internet. Entretanto, eventualmente o endereço do modem do usuário pode mudar, em decorrência de um eventual desligamento do aparelho ou de configurações preestabelecidas pelo provedor. Essa forma de funcionamento impede o usuário de realizar um relacionamento de nome com seu IP pois este irá ser alterado de tempos em tempos. Através do DDNS é possível por exemplo relacionar o domínio thiagogmta.net a um determinado host, mesmo que o endereço IP desse host mude. Podemos citar o No-IP um dos mais famosos provedores de serviços DDNS que disponibiliza seu serviço de forma gratuita até três hosts.

### 6.1.3. Como funciona?

Ao receber uma solicitação de atribuição de endereço IP o servidor DHCP, após atender a requisição, envia uma notificação ao servidor DNS sobre o novo host e seu endereço. O servidor DNS, por sua vez, de posse daquelas informações, atualiza sua base de dados de endereços.

## 6.2. DDNS: Exemplificação

### 6.2.1. Criação do Ambiente



Leve em consideração que nosso ambiente tem a seguinte topologia implementada através do netkit:

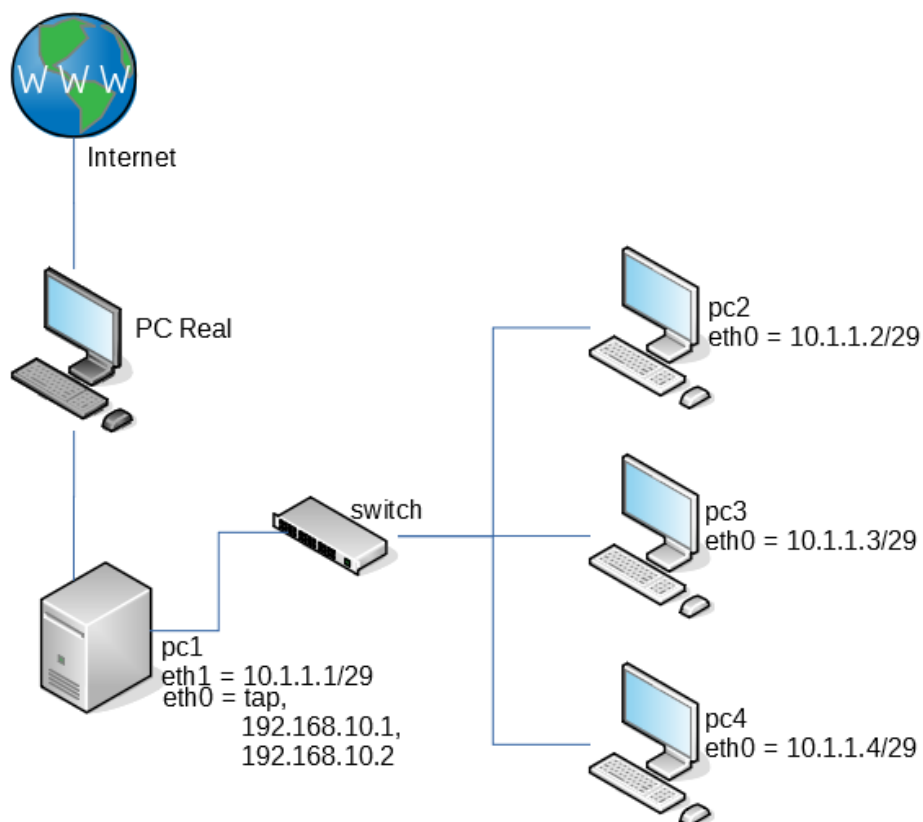


Figura 11. Topologia de testes



Para implementação do DDNS será necessário a configuração de um servidor DHCP e o relacionamento deste servidor com o servidor DNS. Dessa forma, iremos levar em consideração que o PC1 será o servidor DHCP e DNS.

## 6.2.2. Configuração do servidor DHCP

O servidor DHCP deve possuir um IP fixo. Dessa forma este host será configurado com o endereço IP 10.1.1.1.

```
$ ifconfig eth1 10.1.1.1/29
```

### Configurações Iniciais

As configurações referentes ao dhcp serão feitas no arquivo /etc/dhcp3/dhcpd.conf



É possível inserir as configurações desejadas diretamente no arquivo original ou fazer um backup do arquivo e criar um novo apenas com as configurações desejadas. Fica ao seu critério. Para o desenvolvimento deste material foi criado um novo arquivo.

```
$ mv /etc/dhcp3/dhcpd.conf /etc/dhcp3/dhcpd.conf.old  
$ touch /etc/dhcp3/dhcpd.conf  
$ nano /etc/dhcp3/dhcpd.conf
```

No novo arquivo de configuração foram inseridas as seguintes linhas:

```
ddns-update-style none; ①  
default-lease-time 600; ②  
max-lease-time 7200; ③  
authoritative; ④  
  
subnet 10.1.1.0 netmask 255.255.255.248{ ⑤  
    range 10.1.1.2 10.1.1.6; ⑥  
    option subnet-mask 255.255.255.248; ⑦  
    option routers 10.1.1.1; ⑧  
    option broadcast-address 10.1.1.7; ⑨  
    option domain-name-servers 10.1.1.1; ⑩  
    option domain-name "thiago.net"; ⑪  
}
```

- ① Desabilita atualização dinâmica
- ② Informa que o servidor dhcp irá verificar a cada 600 segundos se as estações estão ativas.
- ③ Determina o tempo máximo que um host pode ficar com um endereço.
- ④ Define como servidor DHCP principal da rede
- ⑤ Informa o endereço da rede e sua máscara. Dentro do seu bloco são inseridas as configurações para distribuição de endereços.
- ⑥ Faixa de distribuição de endereços

- ⑦ Máscara da rede
- ⑧ Endereço de Gateway padrão
- ⑨ Endereço de broadcast da rede
- ⑩ Servidor DNS (caso exista)
- ⑪ Nome do domínio

## > Interfaces

Como nosso servidor possui duas interfaces de rede precisamos informar ao servidor qual das duas interfaces irá responder às solicitações de endereçamento IP. Para isso devemos editar o arquivo `/etc/default/dhcp3-server`. Dentro dessa linha alterar o parâmetro `INTERFACES` e informar a interface desejada ficando dessa maneira:

```
#/etc/default/dhcp3-server  
INTERFACES="eth1"
```

Salve e feche o arquivo.

Ao finalizar inicie (ou reinicie) o servidor para validar as configurações.

```
$ /etc/init.d/dhcp3-server start
```

## Testando

Feitas as configurações, acessaremos apenas o host **pc2** inicialmente e solicitaremos atualização de endereço.

```
$ dhclient eth0
```

Posteriormente faremos testes nos pcs 3 e 4.

Caso seja necessário a remoção de endereço em algum dos hosts para realizar novos testes utilize o comando:

```
$ ip address flush dev eth0
```

Configurações iniciais do servidor DHCP finalizadas.

## 6.2.3. Configuração do servidor DNS

Feita a configuração do servidor DHCP partiremos para a configuração do servidor DNS.



Utilizaremos como base o domínio `thiago.net`. Não se esqueça de configurar essa informação no arquivo `/etc/hosts`.

## Criação das Zonas

O primeiro passo para configuração do DNS é acessar o arquivo `named.conf.local` e criar as zonas de configuração.

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

```
//Zona de pesquisa direta
zone "thiago.net"{ ❶
    type master; ❷
    file "/etc/bind/db.thiago.net"; ❸
};

//Zona de pesquisa reversa
zone "1.1.10.in-addr.arpa"{ ❹
    type master;
    file "/etc/bind/db.10"; ❺
};
```

- ❶ Nome da zona onde `thiago.net` é o domínio
- ❷ Tipo de zona
- ❸ Caminho do arquivo da zona direta (deverá ser criado)
- ❹ Zona reversa que é representada pelo Ip do servidor ao contrário sem o ultimo octeto seguido da terminação `.in-addr.arpa` (padrão)
- ❺ Caminho do arquivo de zona reversa

## Configurando os Encaminhadores

A medida que nosso servidor DNS receber requisições de resolução de nomes quais ele não saiba a resposta deverá encaminha-las a um próximo servidor DNS. Descomentar o bloco representado lobo a baixo e inserir um endereço Ip de outro servidor DNS.

```
$ nano /etc/bind/named.conf.options
```

```
fowarders{
    8.8.8.8
}
```

## Criação de arquivos para as Zonas: Direta e reversa

Para facilitar o trabalho faremos uma cópia de dois arquivos já existentes que servirão de base:



```
$ cp /etc/bind/db.local /etc/bind/db.thiago.net
$ cp /etc/bind/db.127 /etc/bind/db.10
```

## Zona Direta

```
$ nano /etc/bind/db.thiago.net
```

```
; BIND - Zona Direta (nome.com.br)
;---
$TTL      604800
@         IN      SOA      pc1.thiago.net. root.thiago.net. (
                                2014051801      ; Serial
                                604800          ; Refresh
                                86400           ; Retry
                                2419200         ; Expire
                                604800 )        ; Negative Cache TTL
;
@         IN      NS       pc1.thiago.net.
pc1       IN      A        10.1.1.1

@         IN      NS       localhost.
@         IN      A        127.0.0.1
@         IN      AAAA     ::1
```

## Zona Reversa

```
$ nano /etc/bind/db.10
```

```
; BIND - Zona Reversa (nome.com.br)
;---
$TTL      604800
@         IN      SOA      pc1.thiago.net. root.thiago.net. (
                                2014051801      ; Serial
                                604800          ; Refresh
                                86400           ; Retry
                                2419200         ; Expire
                                604800 )        ; Negative Cache TTL
;
@         IN      NS       pc1.thiago.net.
1         IN      PTR      pc1.thiago.net.

@         IN      NS       localhost.
1.0.0     IN      PTR      localhost.
```

## Validando configurações

Após as configurações podemos checar as configurações de zonas e verificar eventuais erros através dos comandos:

Para checar as configurações do servidor execute o comando:

```
$ named-checkconf
```

Para verificar as configurações das zonas utilize os comandos:

```
$ named-checkzone thiago.net /etc/bind/db.thiago.net  
$ named-checkzone thiago.net /etc/bind/db.10
```

Caso retorne OK a configuração está correta. Inicie o servidor.

```
$ /etc/init.d/bind start
```

### Testando o servidor DNS

Para verificar se as zonas foram preenchidas corretamente podemos verificar sua tabela através do comando:

```
$ host -l thiago.net
```

Para testar se os hosts cadastrados estão respondendo por seus respectivos nomes podemos utilizar os comandos ping ou host a partir do pc2 para o pc1.

```
$ host pc1.thiago.net  
$ ping pc1.thiago.net
```

O comando host irá retornar o endereço Ip do pc1 e o comando ping irá enviar um pacote para o ip correspondente ao pc1.

## 6.3. Configurando o Dynamic Domain Name System

O ponto interessante do DDNS é que a medida que o servidor DHCP recebe requisições dos clientes ele envia essas informações ao servidor DNS que por sua vez atualiza suas tabelas de resolução de nomes. As configurações apresentadas até este ponto são configurações básicas para o funcionamento de ambos os serviços DHCP e DNS. Faremos então algumas alterações para habilitar o DDNS.

### 6.3.1. Preparando o servidor DHCP para atualizações Dinâmicas

**Criando chave de autenticação**

```
dnssec-keygen -a HMAC-MD5 -b 128 -n HOST ddns
```

Ao executar o comando teremos um retorno semelhante a esse:

```
K<nome da chave>.+157+60250
```

Também será criado um arquivo de mesmo nome. No caso desse exemplo foi criado o arquivo: *Kddns.+157+60250.key*

acesse o arquivo criado copie a chave gerada. No caso desse exemplo a chave gerada foi: *N5+rdUh6jLGHPinu4bmalg==*

acesse o arquivo dhcpd.conf e certifique-se das seguintes configurações:

### Realizando configurações

```
ddns-update-style interim;  
option domain-name "thiago.net";  
authoritative;  
log-facility local7;
```

Crie uma sessão para a chave

```
key ddns {  
    algorithm hmac-md5;  
    secret "<sua chave aqui>";  
};
```

Defina suas zonas

```
zone thiago.net. {  
    primary 10.1.1.1;  
    key ddns;  
}  
  
zone 1.1.10.in-addr.arpa. {  
    primary 10.1.1.1;  
    key ddns;  
}
```

Reinicie o servidor dhcp

## 6.3.2. Preparando o servidor DNS para atualizações Dinâmicas

Acesse o arquivo de configuração named.conf e insira a chave criada no passo anterior:

```
key ddns {
    algorithm hmac-md5;
    secret "<sua chave aqui>";
};
```

Crie a sessão controls no mesmo arquivo named.conf

```
controls {
    inet 127.0.0.1 port 953
    allow { 127.0.0.1; }
    keys { "ddns"; };
};
```

Acesse o arquivo named.conf.local e adicione as seguintes configurações em seus arquivos de zonas:

```
zone "thiago.net" {
    type master;
    file "/etc/bind/db.thiago.net";

    notify no;
    allow-update { key ddns; };
};

zone "1.1.10.in-addr.arpa"{
    type master;
    file "/etc/bind/db.10";

    notify no;
    allow-update { key ddns; };
};
```

Altere o conteúdo do arquivo rndc.key (/etc/bind/rndc.key) e insira a sua chave dentro do arquivo.

Vamos checar as configurações do servidor:

```
$ named-checkconf
```

Caso tudo esteja ok, reinicie o servidor.

## O pulo do Gato

Para máquinas linux é preciso alterar o arquivo */etc/dhcp3/dhclient.conf* nos computadores clientes e alterar a seguinte linha:

```
send host-name "hostname";
```

Onde "hostname" deve ser substituído pelo nome da máquina

## Testando

Chegou a hora da verdade!

1. Execute, no servidor, o comando `tail -f /var/log/syslog` para monitorar as entradas do servidor. Através desse comando poderemos visualizar as solicitações de endereçamento IP dos nossos clientes.
2. Lembra que até o momento só setamos endereço IP no pc 02 através do dhcp? Bom agora vamos fazer o teste nos pcs 03 e 04 execute o comando `dhclient eth0`. == Servidor Web - Apache Prof. Thiago Guimarães Tavares - [thiagogmta@gmail.com](mailto:thiagogmta@gmail.com) IFTO Campus Palmas

### Objetivos do capítulo

Ao final deste capítulo você deverá ser capaz de:

- Entender o que é e para que serve um servidor Web Apache.
- Ser capaz de realizar as configurações iniciais para o funcionamento do servidor como: Iniciar, parar e reiniciar o serviço; Realizar alteração de diretórios e portas de funcionamento do servidor.
- Realizar a configuração de hosts virtuais.

## 6.4. Apresentação

### O que é?

Um servidor web é responsável pela gestão e armazenamento de diversos sites e serviços disponíveis na internet. Desde pequenos sites a grandes canais de entretenimento, redes sociais e mecanismos de busca, são hospedados em servidores web (MORIMOTO, 2013).

### Para que serve?

Um servidor web conta com uma gama de módulos que vão fornecer as funcionalidades necessárias para o funcionamento dos serviços ali hospedados. A exemplo temos uma aplicação comumente conhecida como LAMP que combina alguns dos módulos essenciais para hospedagem de sites. O LAMP é acrônimo para Linux + Apache + MySQL + PHP. (MORIMOTO, 2013).

### Como funciona?

Quando o usuário digita o endereço de um determinado site o servidor DNS converte esse nome em um endereço IP. Esse endereço é correspondente ao endereço do servidor que hospeda o site. Ao chegar no servidor a requisição é tratada de forma a retornar para o browser do usuário a página desejada. (MORIMOTO, 2013). Nesta sessão sobre servidores web nos atentaremos a respeito da configuração do Apache.

## 6.5. Apache: Considerações Iniciais

**Instalação** O Apache2 pode ser instalado através do seguinte comando:

```
$ sudo apt-get install apache2
```

No netkit o apache já vem instalado por padrão. Ao instalar o apache2, também é instalado o suporte ao SSL automaticamente.

Iniciar, parar e reiniciar o servidor:

```
$ /etc/init.d/apache2 start  
$ /etc/init.d/apache2 stop  
$ /etc/init.d/apache2 restart
```

### Arquivos de configuração

Todos os arquivos de configuração do servidor Web Apache estão disponíveis em /etc/apache2. Segue sua descrição:

- apache2.conf - Arquivo de configuração principal do apach
- mods-available - Módulos disponíveis no servidor
- mods-enabled - Módulos ativos no servidor
- ports.conf - Arquivo de configuração para alteração de portas no servidor
- sites-available - Diretório que armazena os hosts virtuais no servidor
- sites-enabled - Diretório que armazena os hosts virtuais ativos no servidor

### Comandos

Existem alguns comandos que podem ser utilizados para habilitar ou desabilitar um módulo ou um host virtual:

- a2ensite - Habilita um host virtual
- a2dissite - Desabilita um host virtual
- a2enmod - Habilita um módulo específico
- a2dismod - Desabilita um módulo específico

Ao trabalharmos com hosts virtuais esses comandos serão melhor detalhados.

## 6.6. Minha primeira hospedagem

Por padrão nosso servidor já vem com uma pequena página de testes que pode ser visualizada no navegador assim que o servidor for iniciado. Vamos dar uma olhada. No seu servidor apache inicie o serviço com o seguinte comando:

```
$ /etc/init.d/apache2 start
```

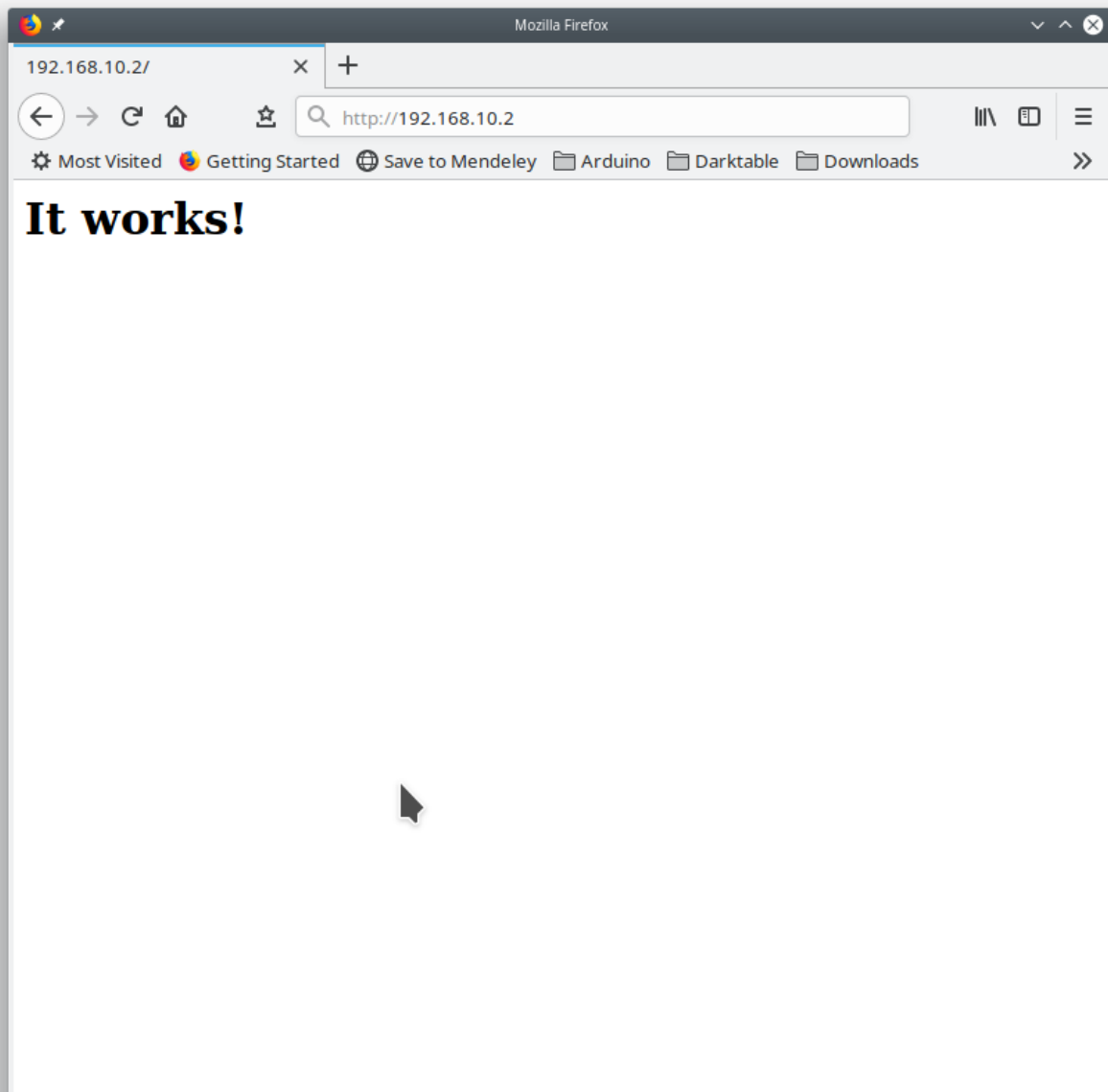
Lembrando que nenhuma configuração foi feita no servidor, apenas inicializado o serviço, você deverá visualizar uma tela semelhante a esta:

A terminal window titled 'pc1' with standard window controls. The terminal shows the command '/etc/init.d/apache2 start' being executed. The output is: 'Starting web server: apache2' followed by a warning message: 'apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for ServerName'. The prompt returns to 'pc1:~#'.

```
pc1:~# /etc/init.d/apache2 start
Starting web server: apache2
apache2: Could not reliably determine the server's fully
qualified domain name, using 127.0.0.1 for ServerName
pc1:~#
```

Figura 12. *apache2 start*

Podemos testar o funcionamento do nosso servidor utilizando qualquer navegador de internet de nosso host anfitrião (Pc Real) bastando apenas inserir no navegador de internet o endereço IP da nossa conexão **tap**.



*Figura 13. It Works!*

O apache apresenta essa tela de testes por padrão.

Faremos uma pequena modificação a nível de teste. Vamos criar um novo usuário em nosso servidor, criar um diretório para armazenar uma página teste dentro do /home desse usuário e apontar nosso servidor Apache para esse diretório. Are you ready?

Para criar um novo usuário utilizamos o seguinte comando:

```
$ adduser pedro
```

Serão solicitadas informações do usuário como nome e telefone. Preencha as que forem pertinentes e confirme com Y ao final. Ao criar um novo usuário é criado também um novo diretório em /home nesse caso: /home/pedro. Dentro desse diretório criaremos uma pasta chamada "www" para armazenar nossos arquivos da nova página teste e um arquivo chamado index.html que será nossa página inicial.



```
$ cd /home/pedro
$ mkdir www
$ cd www
$ touch index.html
```

O index.html receberá o seguinte conteúdo:

```
<html>
  <head>
    <title>Site do Pedro</title>
    <meta charset="utf8">
  </head>
  <body>
    <h1>Bem vindo ao site do Pedro!</h1>
    <hr/>
    <p> Olá seja bem vindo! </p>
  </body>
</html>
```

Pronto! Agora precisamos apontar nosso diretório default do servidor apache para esse novo diretório. Para tanto devemos editar o arquivo **default** em */etc/apache2/sites-available*. Altere as linhas:

```
DocumentRoot e Directory
```

Para ambas as linhas, substituir o caminho padrão (/var/www) pelo novo caminho: /home/pedro/www. Reinicie o servidor e atualize a página do navegador. Você deverá obter um resultado semelhante a este:

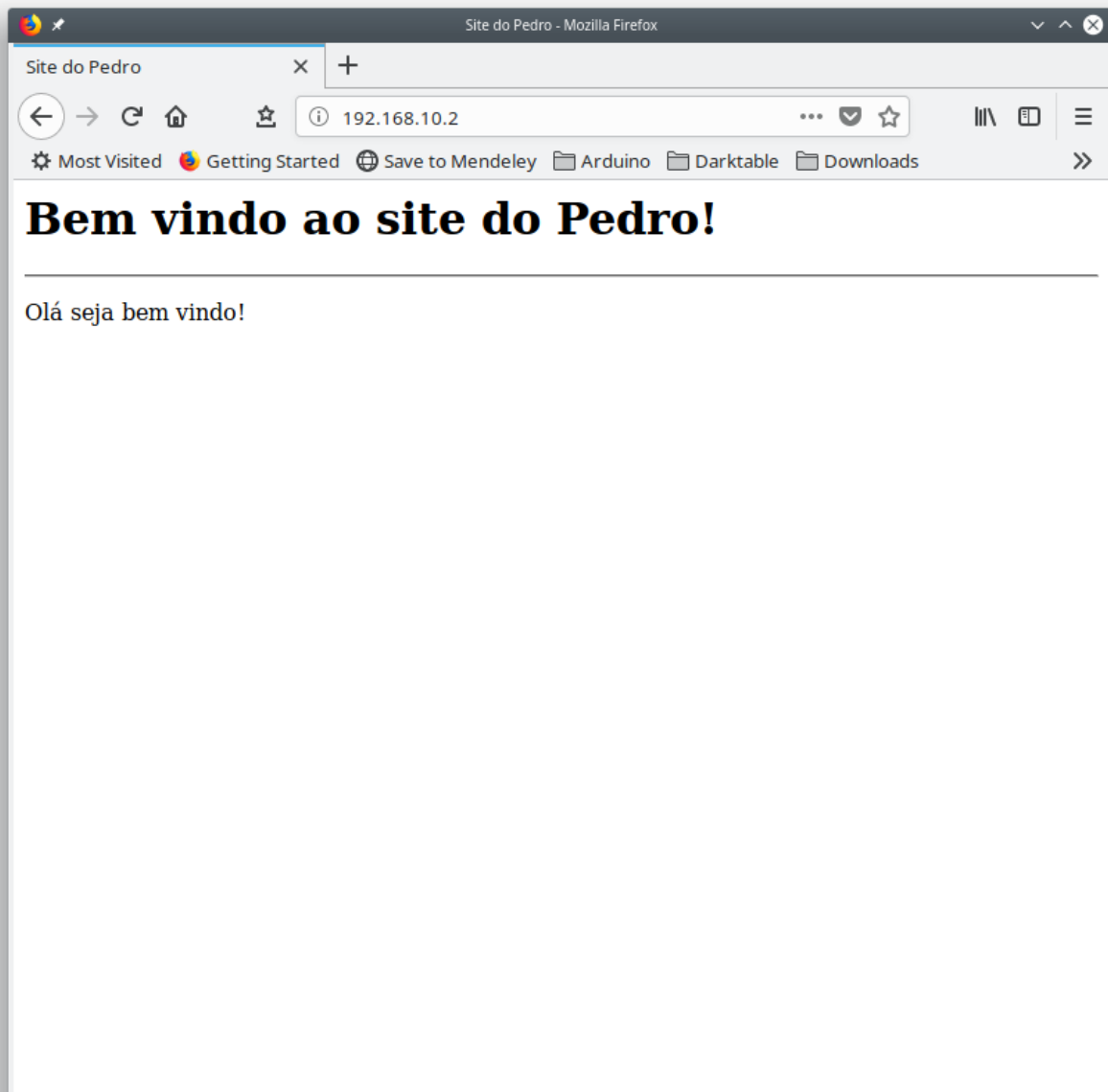


Figura 14. Site do Pedro

### Alterando a porta padrão do Apache

Por padrão o apache funciona através da porta 80. Caso necessário essa configuração pode ser alterada. Para alterar a porta padrão do apache deve-se alterar o arquivo **ports.conf** localizado em */etc/apache2*. Ao acessar esse arquivo altere as linhas:

*O número 80 informa a porta de funcionamento. Caso queira que o apache opere em outra porta basta alterar essas duas linhas.*

```
NameVirtualHost *:80
Listen 80
```

Posteriormente deve-se alterar o arquivo responsável pelo host, no nosso caso o arquivo **default** localizado em */etc/apache2/sites-available*. Nesse arquivo localize a linha

Altere a porta conforme feito no arquivo `ports.conf`

```
<VirtualHost *:80>
```



Não se esqueça de reiniciar o servidor!

Para verificar as novas configurações acesse pelo navegador da seguinte maneira:

*Lembre-se que 192.168.10.2 é nosso endereço ip entre o computador e o netkit. A informação 8080 diz respeito a nova porta que foi inserida (como exemplo).*

```
http://192.168.10.2:8080
```

## 6.7. Hosts Virtuais

### Definição:

Nosso servidor web é capaz de suportar vários sites hospedados fazendo distinção entre eles de acordo com as requisições dos usuários. A medida que o usuário envia um pacote de solicitação de acesso à um determinado site hospedado no servidor, este verifica em sua base se tem alguma hospedagem com aquele domínio. Em caso positivo o servidor irá responder a solicitação apontando para os arquivos do site correspondente à solicitação. Para que isso funcione é necessário que cada domínio tenha seu arquivo de configuração de forma a apontar para os arquivos daquele site. A imagem a baixo representa o funcionamento.

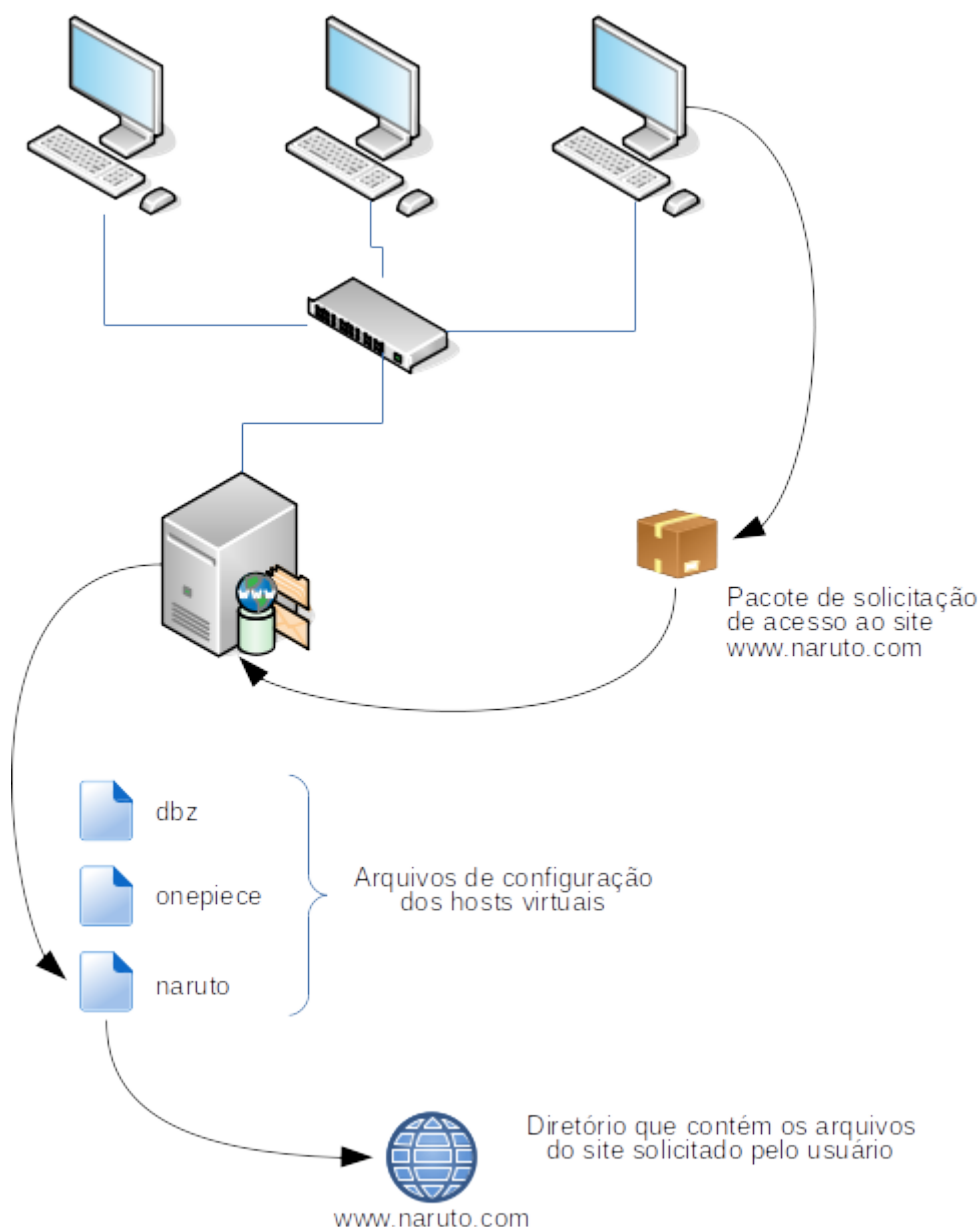


Figura 15. Representação do funcionamento dos hosts virtuais no apache

### Criando estrutura

Vamos levar em consideração que nosso servidor irá hospedar os seguintes três sites:

- [www.dbz.com](http://www.dbz.com)
- [www.naruto.com](http://www.naruto.com)
- [www.onepiece.com](http://www.onepiece.com)

Para melhor organização vamos criar um usuário para cada host virtual e armazenar o conteúdo de cada site dentro da pasta do usuário. Imagine que cada usuário poderá posteriormente ter acesso ao seu diretório com seu usuário e senha para fazer upload ou alterar os arquivos do seu próprio site. Os passos a seguir foram executados no servidor web apache com o usuário root.

*Criando usuários (preencha as informações pertinentes a cada usuário e confirme com y).*

```
$ adduser dbz
$ adduser naruto
$ adduser onepiece
```

Usuários criados, vamos criar o diretório `www` dentro do `/home` de cada usuário, criar um arquivo `index.html` para cada um e alterar as configurações de propriedade dos arquivos:

*Criando estrutura de diretórios e arquivos `index.html`*

```
$ mkdir /home/dbz/www
$ mkdir /home/naruto/www
$ mkdir /home/onepiece/www

$ touch /home/dbz/www/index.html
$ touch /home/naruto/www/index.html
$ touch /home/onepiece/www/index.html
```

Vamos agora alterar as configurações de propriedade dos arquivos.



Como os diretórios e arquivos foram criados com o usuário `root`. O usuário `root` é dessa forma proprietário dos diretórios e arquivos. Vamos alterar as configurações de propriedade dos arquivos para que cada usuário tenha permissão de escrita em seu próprio diretório `www`.

*Alterando propriedades*

```
$ chown -R dbz:dbz /home/dbz/www
$ chown -R naruto:naruto /home/naruto/www
$ chown -R onepiece:onepiece /home/onepiece/www
```

Estrutura pronta, vamos apenas alterar o conteúdo do arquivo `index.html` de cada site:

*`index.html`: DBZ*

```
<html>
  <head>
    <title>Dragon Ball Z</title>
    <meta charset="utf8">
  </head>
  <body>
    <h1>Posso pressentir o perigo e o caos...</h1>
    <hr/>
    <p> "Você não é derrotado quando perde, mais sim quando você desiste."
(VEGETA). </p>
  </body>
</html>
```

*index.html: NARUTO*

```
<html>
  <head>
    <title>Naruto</title>
    <meta charset="utf8">
  </head>
  <body>
    <h1>Vila Oculta da Folha</h1>
    <hr/>
    <p> "Se você não gosta do seu destino, não aceite. Em vez disso, tenha a
coragem de o mudar do jeito que você quer que seja.." (UZUMAKI NARUTO). </p>
  </body>
</html>
```

*index.html: ONEPIECE*

```
<html>
  <head>
    <title>Naruto</title>
    <meta charset="utf8">
  </head>
  <body>
    <h1>Akuma No Mi</h1>
    <hr/>
    <p> "Não importa o que o mundo diz de mim, o que importa é que eu nunca fiz
nada que contrariasse os meus princípios e nunca farei." (RORONOA ZORO). </p>
  </body>
</html>
```

## Criando arquivos de configuração dos Hosts Virtuais

Os arquivos de virtual host devem ficar armazenados no diretório */etc/apache2/sites-available* e especifica como o servidor apache irá responder às solicitações daquele domínio em específico. Nesse diretório encontra-se o arquivo **default** que pode ser utilizado de base para os demais arquivos de hosts virtuais ou podemos criar um arquivo em branco e inserir as configurações mínimas necessárias. Essa segunda opção é o que faremos.

*Criando arquivos dos Hosts Virtuais*

```
$ cd /etc/apache2/sites-available
$ touch dbz
$ touch naruto
$ touch onepiece
```

Faremos a configuração do Virtual Host **dbz**. Para os demais Hosts Virtuais, aproveite esta mesma configuração alterando apenas o necessário:

## Configuração do host dbz

```
<VirtualHost *:80>
  ServerAdmin admin@dbz.com ①
  ServerName dbz.com ②
  ServerAlias www.dbz.com ③
  DocumentRoot /home/dbz/www ④
  ErrorLog /etc/apache2/error.log ⑤
  CustomLog /etc/apache2/access.log combined ⑥
</VirtualHost>
```

- ① E-mail do administrador do site.
- ② Domínio base do virtual host.
- ③ Outros nomes que podem corresponder ao Virtual Host.
- ④ Localização dos arquivos do domínio em questão
- ⑤ Caminho para o arquivo de log de erro
- ⑥ Caminho para o arquivo de log de requisições ao servidor




Para os demais Hosts Virtuais a estrutura é a mesma.

Nosso host virtual foi criado entretanto ainda não está ativo. Para ativar nosso host em questão basta utilizar o comando a seguir:

### Ativando o site dbz

```
$ a2ensite dbz
```

Deve retrornar uma mensagem parecida com esta:

A terminal window titled 'pc1' with a dark header bar. The terminal shows the command 'a2ensite dbz' being executed in the directory 'pc1:/etc/apache2/sites-available#'. The output indicates that the site 'dbz' is being enabled and provides instructions to reload the Apache2 service. The prompt returns to 'pc1:/etc/apache2/sites-available#' with a cursor.

```
pc1:/etc/apache2/sites-available# a2ensite dbz
Enabling site dbz.
Run '/etc/init.d/apache2 reload' to activate new configuration!
pc1:/etc/apache2/sites-available#
```

Figura 16. *a2ensite*

A mensagem informa que o servidor deve ser atualizado. Note que o arquivo de configuração do novo virtual host também irá aparecer agora no diretório */etc/apache2/sites-enabled*. Isso significa que o site está ativo.

Reinicie o servidor!

## Testando os hosts virtuais

### Configuração dos clientes

Como não dispomos de um servidor DNS para fazer a conversão de nomes (até este ponto, pois isso será feito no próximo capítulo) iremos modificar temporariamente o arquivo **hosts** para realizar testes locais. Esse passo deve ser feito nos computadores clientes (pc2, pc3 e pc4). Para tanto acesse o arquivo */etc/hosts* e adicione as seguintes linhas:

*/etc/hosts*

```
10.1.1.1 www.dbz.com
10.1.1.1 www.naruto.com
10.1.1.1 www.onepiece.com
```

### Teste

Nossos testes de hosts virtuais serão executados apenas em nosso ambiente de testes Netkit através de nossos clientes. Como o netkit não trabalha com interface gráfica iremos utilizar um navegador de internet em modo texto nos clientes para validar nosso resultado. O netkit já possui o **Links** que



é um navegador de internet em modo texto. O Links já oferece funcionalidade suficiente para testarmos nosso ambiente. Basta apenas, no terminal do pc2 por exemplo, executar o comando:

Acesso ao site dbz pelo Pc2

```
$ links www.dbz.com
```

A imagem a seguir ilustra o acesso aos três sites criados através dos pcs 2, 3 e 4 utilizando o comando **links**.

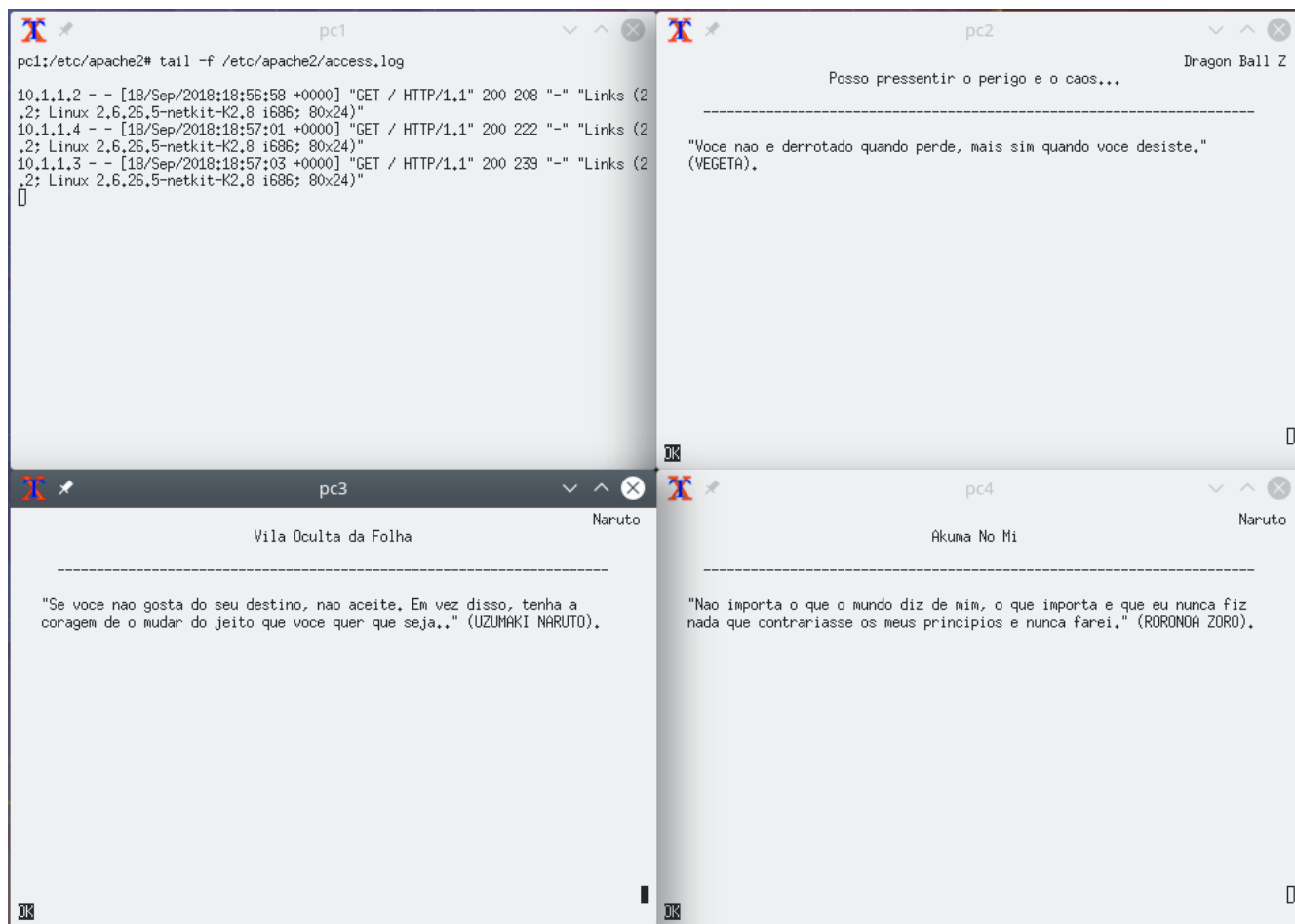


Figura 17. Teste de acesso aos Hosts Virtuais



No terminal do pc1 foi executado o comando: `tail -f /etc/apache2/access.log`. Esse comando analisa as ultimas entradas no arquivo `access.log` que por sua vez acusa acessos ao nosso servidor. Observando atentamente podemos perceber que o nosso servidor recebeu três solicitações provenientes dos endereços: 10.1.1.2, 10.1.1.3 e 10.1.1.4 pertencentes aos pcs 2, 3 e 4.

## 6.8. Problematização

Seguem dois problemas para testar o que foi absorvido até o momento. O primeiro problema aborda o tópico inicial sobre o servidor web apache, capítulo *Minha Primeira Hospedagem*. O segundo problema aborda o conteúdo apresentado no capítulo *Hosts Virtuais*.

### 1. Minha Primeira Hospedagem

- a. Crie um usuário com seu nome no servidor
- b. Dentro do diretório do seu usuário crie o diretório www
- c. Crie um arquivo index.html com o seguinte conteúdo:

```
h1 - Seu nome
h2 - Nome da Disciplina
h2 - Nome do Professor
p - Um parágrafo sobre o que acha do curso
p - Um parágrafo sobre seu desenvolvimento na disciplina
p - Um parágrafo sobre o que pretende fazer ao final do curso
* Uma imagem qualquer
* Um link para o site do IFTO
h3 - Data
```

## 2. Hosts Virtuais

- a. Crie os três usuários dos hosts virtuais apresentados neste material.
- b. Crie a estrutura de diretórios e os arquivos html de cada host.
- c. Crie as entradas no servidor apache para cada host.
- d. Faça o teste de acesso a cada um dos sites, monitorando o arquivo access.log conforme apresentado no exemplo.