



INSTITUTO FEDERAL

Norte de Minas Gerais

Campus Januária

Campus Party Brasil 15

- Kathará -



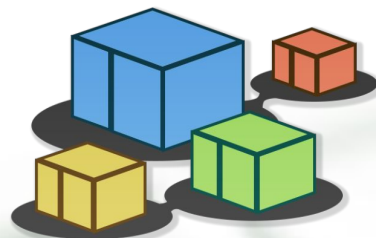
Apresentação Pessoal

■ Prof. Adriano Antunes Prates

- @adrianoantunesp
- <https://github.com/adrianoifnmg>



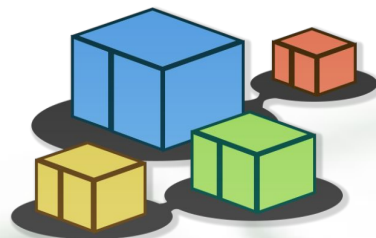
- Bacharel em Sistemas de Informação (Unimontes, 2008)
- Especialista Redes de Computadores (ESAB, 2011)
- Mestre em Computação (UFF, 2014)
 - Redes de Computadores e Sistemas Distribuídos
- Docente do IFNMG desde 2009.
 - Bacharelado em Sistemas de Informação



- **Kathará** é um ambiente *open-source* para **emulação** de redes de computadores baseado na tecnologia de *containers* (docker).
 - *Kathará é uma evolução do projeto **Netkit**.*



Kathará



Kathará

- A ferramenta permite a **criação, configuração e gerenciamento de redes e serviços**, desde as mais simples às mais complexas.

- **Site**

- <https://www.kathara.org/>



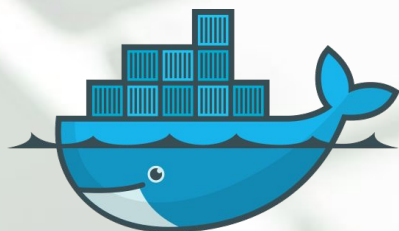


Instalação & Configuração

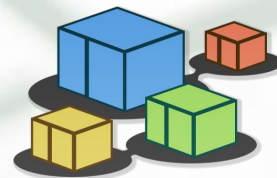
■ Orientações para Instalação e Configuração:

1º Passo - Docker:

```
$ sudo apt-get update  
$ sudo apt-get install docker.io  
$ sudo docker run hello-world
```



docker



Kathará



Instalação & Configuração

■ Or

1º Pa

\$ sudo

\$ sudo

\$ sudo

```
adriano@adriano: ~  
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda  
[sudo] senha para adriano:  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
   (amd64)  
3. The Docker daemon created a new container from that image which runs the  
   executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
   to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/  
adriano@adriano:~$
```



rá



Instalação & Configuração

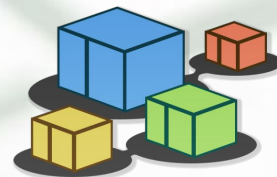
■ Orientações para Instalação e Configuração:

2º Passo - Kathará:

```
$ sudo add-apt-repository ppa:katharaframework/kathara  
$ sudo apt-get update  
$ sudo apt-get install kathara  
$ kathara check
```



docker



Kathará



Instalação & Configuração

■ O

2º P

\$ suc
\$ suc
\$ suc
\$ kat

```
adriano@adriano: ~  
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda  
adriano@adriano:~$ kathara check  
* Current Manager is: Docker (Kathara)  
* Manager version is: 20.10.12  
* Python version is: 3.9.12 (main, Apr 18 2022, 22:40:46) - [GCC 9.4.0]  
* Kathara version is: 3.5.0  
* Operating System version is: Linux-5.4.0-122-generic-x86_64  
* Trying to run 'Hello World' container...  
Deploying devices...|#####| 1/1  
* Container run successfully.  
Deleting devices...|#####| 1/1  
adriano@adriano:~$
```

hara

á



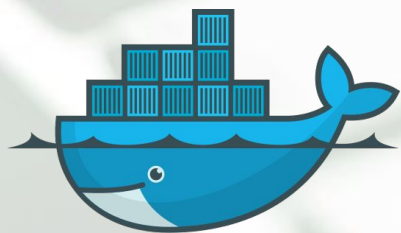


Instalação & Configuração

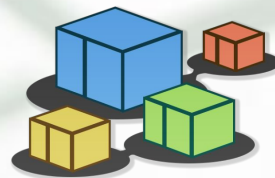
■ Orientações para Instalação e Configuração:

3º Passo - Imagem Customizada e Outras Definições:

```
$ sudo docker pull adrianoantunesp/admin-redes  
$ kathara settings  
$ Opção 2 -> 6: adrianoantunesp/admin-redes  
$ Opção 5 -> 3: /usr/bin/gnome-terminal  
$ Opção 10 -> 1.
```



docker



Kathará



Criando um *Host*

- Criando *Host* contendo uma Interface de Rede

```
$ kathara vstart -n pc0 --eth 0:A
```



Criando um *Host*

■ Criar

```
root@pc0: /
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
root@pc0:/#
```

\$ katha



Criando um *Host*

- Criando *Host* contendo uma Interface de Rede

```
$ kathara vstart -n pc0 --eth 0:A
```

Cria e executa um novo contêiner que representa um host...



Criando um *Host*

- Criando *Host* contendo uma Interface de Rede

```
$ kathara vstart -n pc0 --eth 0:A
```

...nomeado como
“pc0”



Criando um *Host*

- Criando *Host* contendo uma Interface de Rede

```
$ kathara vstart -n pc0 --eth 0:A
```

...e que possui 1 interface de rede padrão Ethernet “eth0” conectada ao domínio de colisão (enlace) “A”



Laboratório 01

- Crie o seguinte laboratório no Kathará:



- Como testar a **conectividade** entre as máquinas?



Revisão Conceitual

Modelo OSI

Camada de Aplicação

Camada de Apresentação

Camada de Sessão

Camada de Transporte

Camada de Rede

Camada de Enlace

Camada de Física

Arquitetura TCP / IP

Camada de Aplicação

Camada de Transporte

Camada Internet / Inter-Redes

Camada Host / Rede ou
Interface de Rede



Interfaces de Rede

- **ifconfig** é a tradicional ferramenta para visualização e configuração das interfaces de rede em plataformas Linux.

```
$ ifconfig
```

- Outro utilitário (**mais recente**) que também permite a visualização e configuração de interfaces é o “**ip**”.

```
$ ip a
```




Configurando uma Interface

```
$ ifconfig eth0 x.y.z.w/z
```

- O comando acima realiza a configuração **temporária** da interface.
 - Atribui o endereço **x.y.z.w** à interface eth0.
 - Atribui máscara de rede correspondente à **/z**.

Obs: Essa configuração é VOLÁTIL.



Finalizando os Hosts

- Para finalizar a execução do laboratório virtual (finalizar todos os contêineres), execute:

```
$ kathara wipe
```

- Para finalizar a execução apenas de um host específico, execute:

```
$ kathara vclean -n cliente
```



Monitoramento de Rede

- Às vezes é tarefa bastante complexa identificar o ponto focal de um problema de comunicação em rede...
 - O problema é na origem ou no destino?
 - Os pacotes estão saindo pela interface?
 - Os pacotes estão chegando no destino?
 - Os pacotes estão se perdendo no meio do caminho?
 - etc...
- Para auxiliar nessa tarefa, utilizamos **ferramentas de monitoramento de rede**, que permitem a captura e análise de pacotes.



TCPDUMP

TCPDUMP
é uma
ferramenta
nativa de
monitoramento
nos sistemas
Linux.

```
root@pc2: /
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
root@pc2:/# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
16:41:07.113067 ARP, Request who-has 192.168.1.2 tell 192.168.1.1, length 28
16:41:07.113114 ARP, Reply 192.168.1.2 is-at 52:8b:35:89:16:b7 (oui Unknown), length 28
16:41:07.113158 IP 192.168.1.1 > 192.168.1.2: ICMP echo request, id 37, seq 1, length 64
16:41:07.113190 IP 192.168.1.2 > 192.168.1.1: ICMP echo reply, id 37, seq 1, length 64
16:41:08.131090 IP 192.168.1.1 > 192.168.1.2: ICMP echo request, id 37, seq 2, length 64
16:41:08.131139 IP 192.168.1.2 > 192.168.1.1: ICMP echo reply, id 37, seq 2, length 64
16:41:09.155052 IP 192.168.1.1 > 192.168.1.2: ICMP echo request, id 37, seq 3, length 64
16:41:09.155087 IP 192.168.1.2 > 192.168.1.1: ICMP echo reply, id 37, seq 3, length 64
16:41:12.258914 ARP, Request who-has 192.168.1.1 tell 192.168.1.2, length 28
16:41:12.258946 ARP, Reply 192.168.1.1 is-at ea:18:99:65:ed:17 (oui Unknown), length 28
^C
10 packets captured
10 packets received by filter
```



TCPDUMP

■ Principais parâmetros de filtragem do TCPDUMP

```
$ tcpdump -i eth0
```

```
$ tcpdump icmp
```

```
$ tcpdump icmp or tcp
```

```
$ tcpdump port 80
```

```
$ tcpdump src port 1025
```

```
$ tcpdump portrange 21-23
```

```
$ tcpdump host ifnmg.edu.br and port https
```

```
$ tcpdump -nX
```

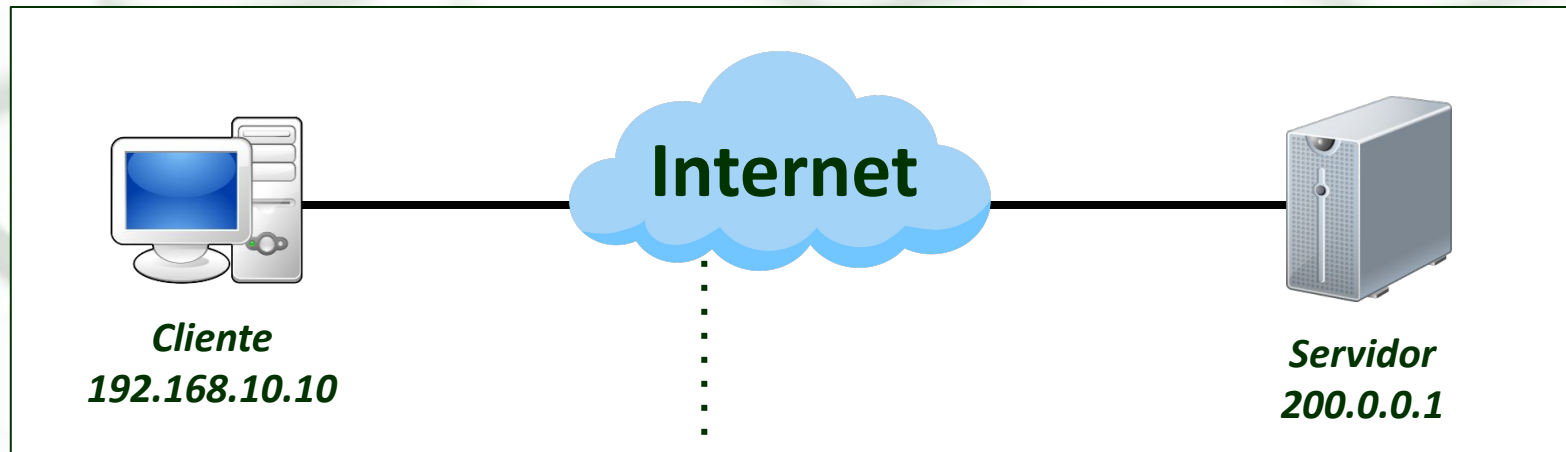
```
$ tcpdump -w arquivoCaptura.pcap
```

[Veja Mais...](#)



Laboratório 02

■ Porque não devemos navegar por sites inseguros?

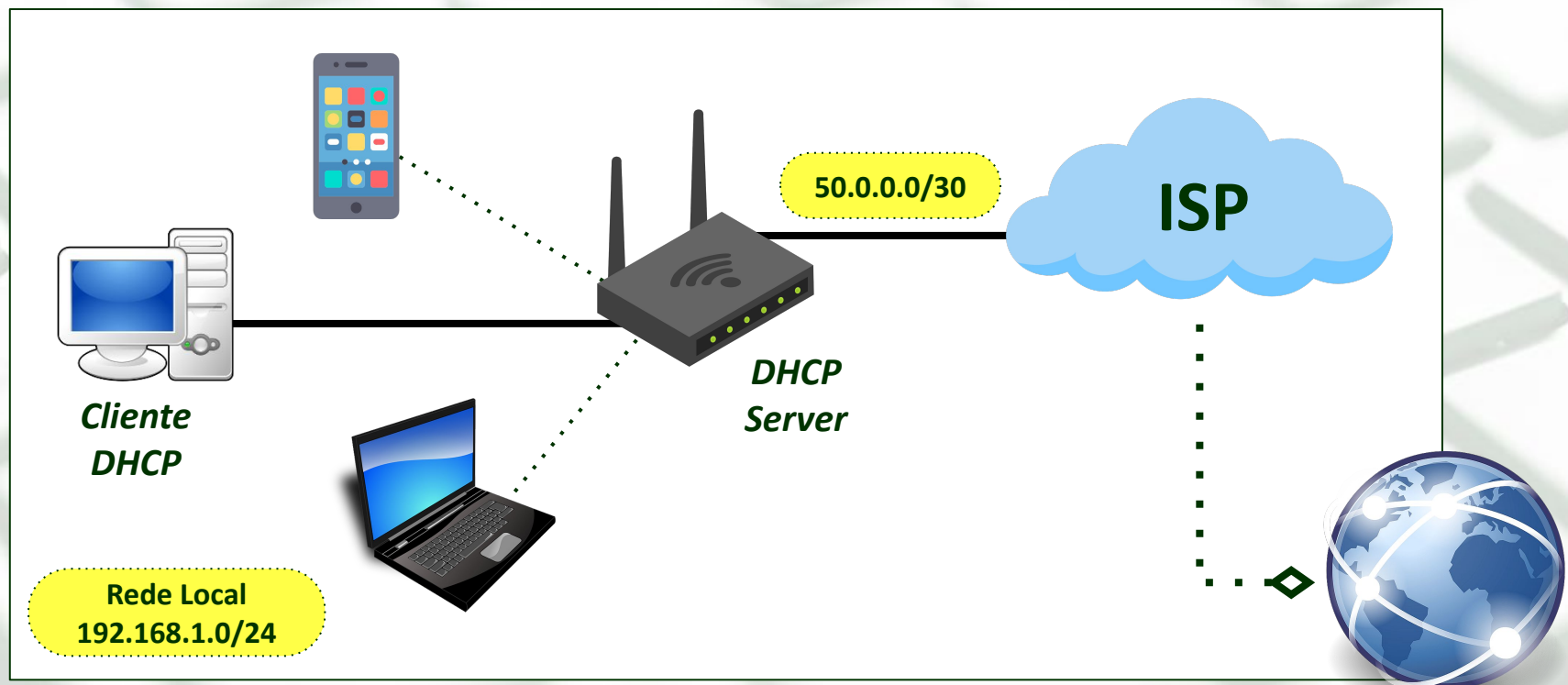


```
$ tcpdump -w escuta.pcap
```



Laboratório 03

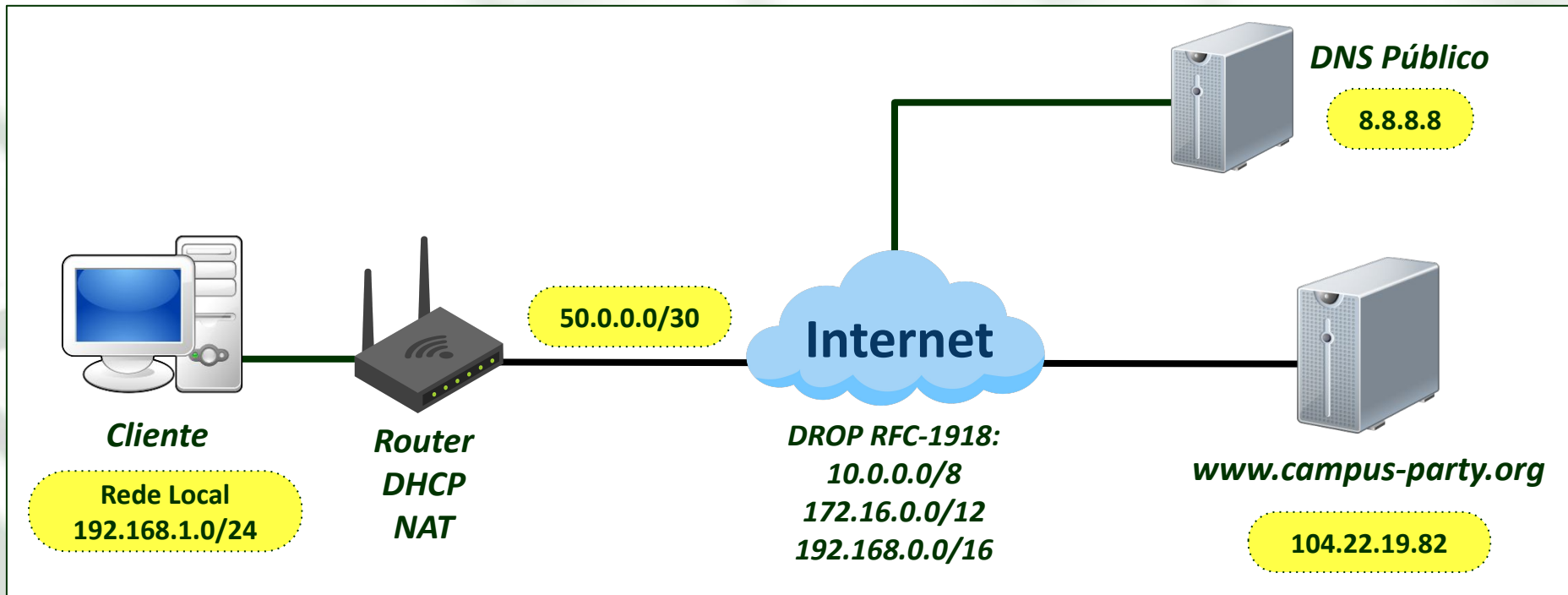
■ Aprendendo sobre Protocolos de Redes (p.Ex.: DHCP) e NAT.





Laboratório 04

■ “Mini Internet”





INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

GRATIDÃO!

A person with dark curly hair, wearing a blue long-sleeved shirt, is holding a white rectangular sign with both hands. The sign has the word "Obrigado" written on it in a black, rounded, sans-serif font. The background is dark and out of focus.

Obrigado