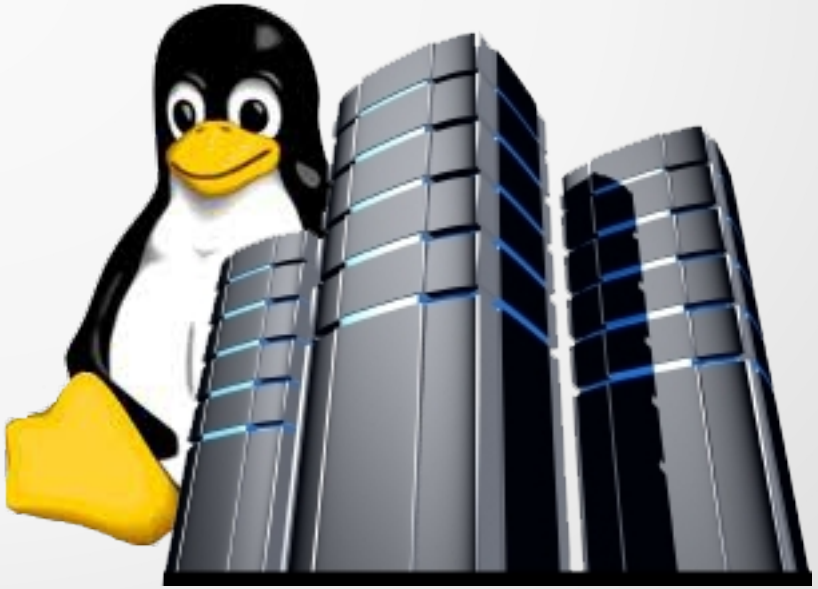
Criando um VPS seguro para suas aplicações PHP



Criando um VPS seguro para suas aplicações PHP

Por **Yves Toupe**

PHP Zend Certified Engineer

www.yvestoupe.com

VPS : Significado

- *virtual private server* (*VPS*)
- *virtual dedicated server* (*VDS*)

Método de particionamento de um servidor em vários servidores virtuais independentes, cada um com as características de um servidor dedicado , utilizando-se técnicas de virtualização. Cada servidor pode operar com um sistema operacional diferente e reiniciar de forma independente.

VPS : Princípios de arquitectura

Um servidor virtual é uma sub-parte lógica de um servidor de hospedagem. Os recursos são compartilhados entre partições lógicas diferentes que são independentes um do outro.

VPS : Princípios de arquitectura

VPS ou VDS

Um VPS se comporta individualmete como um servidor dedicado de padrão com algumas reservas:

- no nível do kernel (nucleo)
- no sistema de arquivos
- ou na interfaces de rede.

VPS : Princípios de arquitectura

VPS ou VDS

Do ponto de vista do sistema, o servidor virtual é uma máquina virtual. Existem várias soluções de virtualização de um servidor:

- virtualização baseada no princípio do isolamento
- virtualização baseada no princípio do paravirtualização.

VPS : Princípios de arquitectura

Princípio do isolamento

- Uso dum núcleo comum : como OpenVZ ou lxc (Linux Containers).
- oferece um bom desempenho
- mas limita a escolha dos sistema operacional para uso das distribuições Linux

VPS : Princípios de arquitectura

Princípio paravirtualização.

- Funciona como emuladores de hardware: ex: XEN
- cada máquina virtual pode rodar qualquer sistema operacional

VPS : Princípios de arquitectura

VPS ou VDS

- Soluções baseadas no isolamento são freqüentemente chamados de servidores virtuais privadas VPS

- Servidor dedicado virtual VDS é então atribuído para as soluções baseadas paravirtualização,

VPS : Princípios de arquitectura

VPS ou VDS

Mas hoje os nomes comerciais são todos misturados as empresas de hospedagem nao formecem esses detalhes

VPS : Vantagens

para a empresa de hospedagem

- As principais razões para particionar uma máquina física em múltiplos servidores virtuais sao para melhorar a segurança:
  - Reduz o custo e o número de servidores físicos necessários

VPS : Vantagens

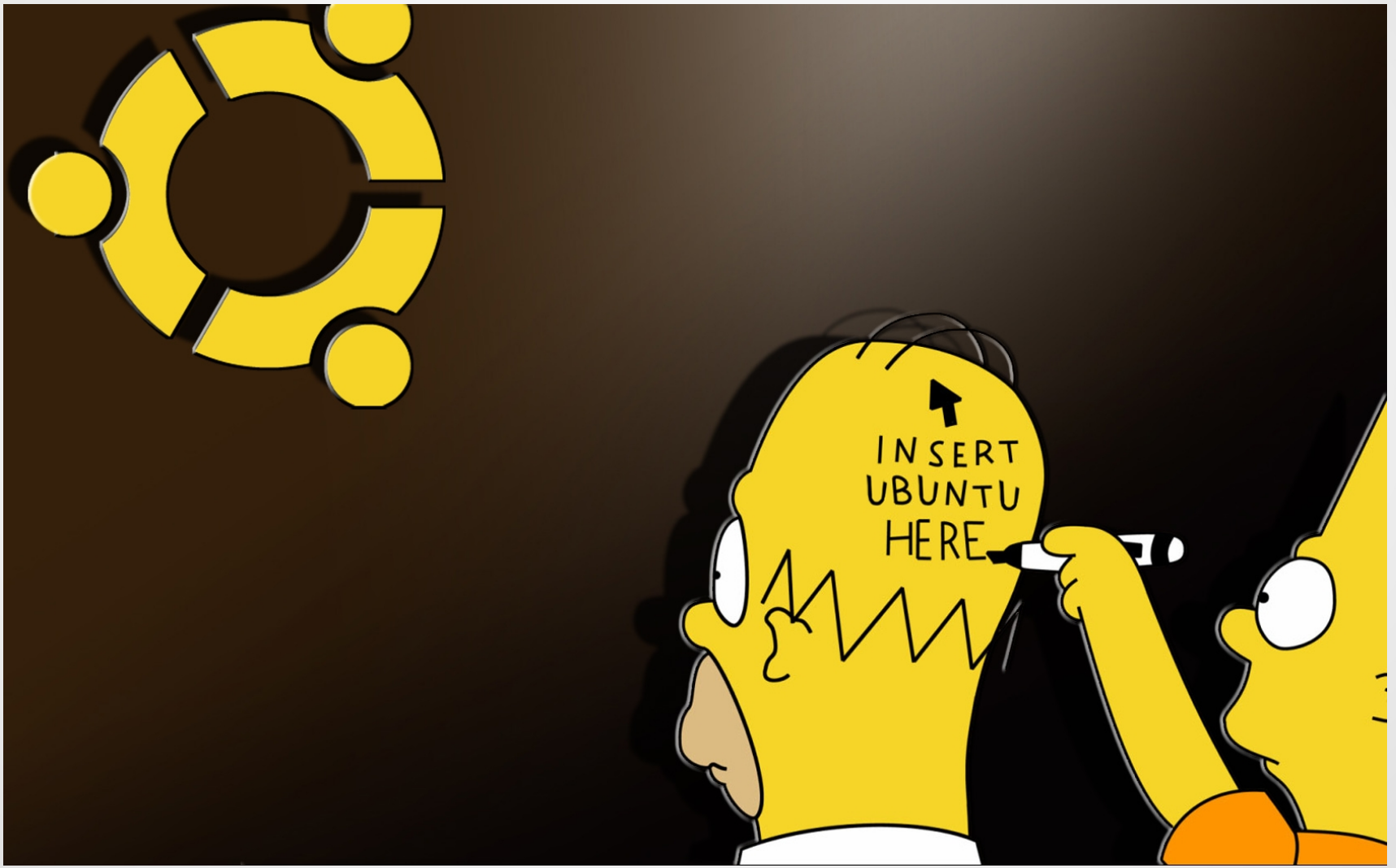
para o cliente

- configurar o servidor de acordo com de acordo com as necessidades do seu aplicativo
- Escolher o desempenho do servidor
- Recursos: Cada usuário pode ter acesso a todos os recursos e espaço do servidor .
- Baixo custo: só paga pelo uso
- Segurança dos seus applicativos

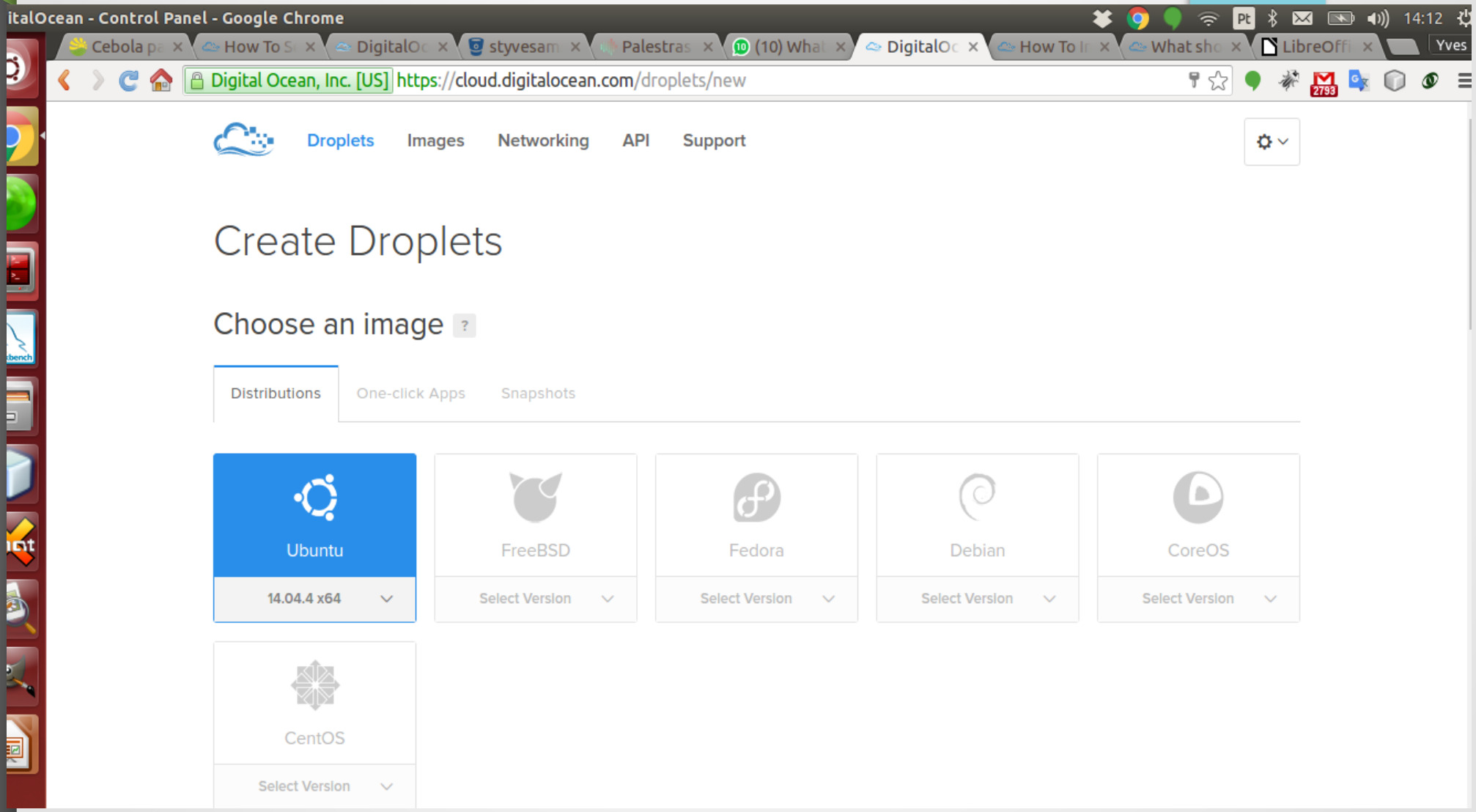
VPS : Uso

- VPS ou VDS são amplamente utilizados para hospedagem na web,
- área da formação, eles são usados para acomodar o ambiente específico para cada aluno.
- Eles também são as vezes usados para hacking ex: ataque DOS/DDOS

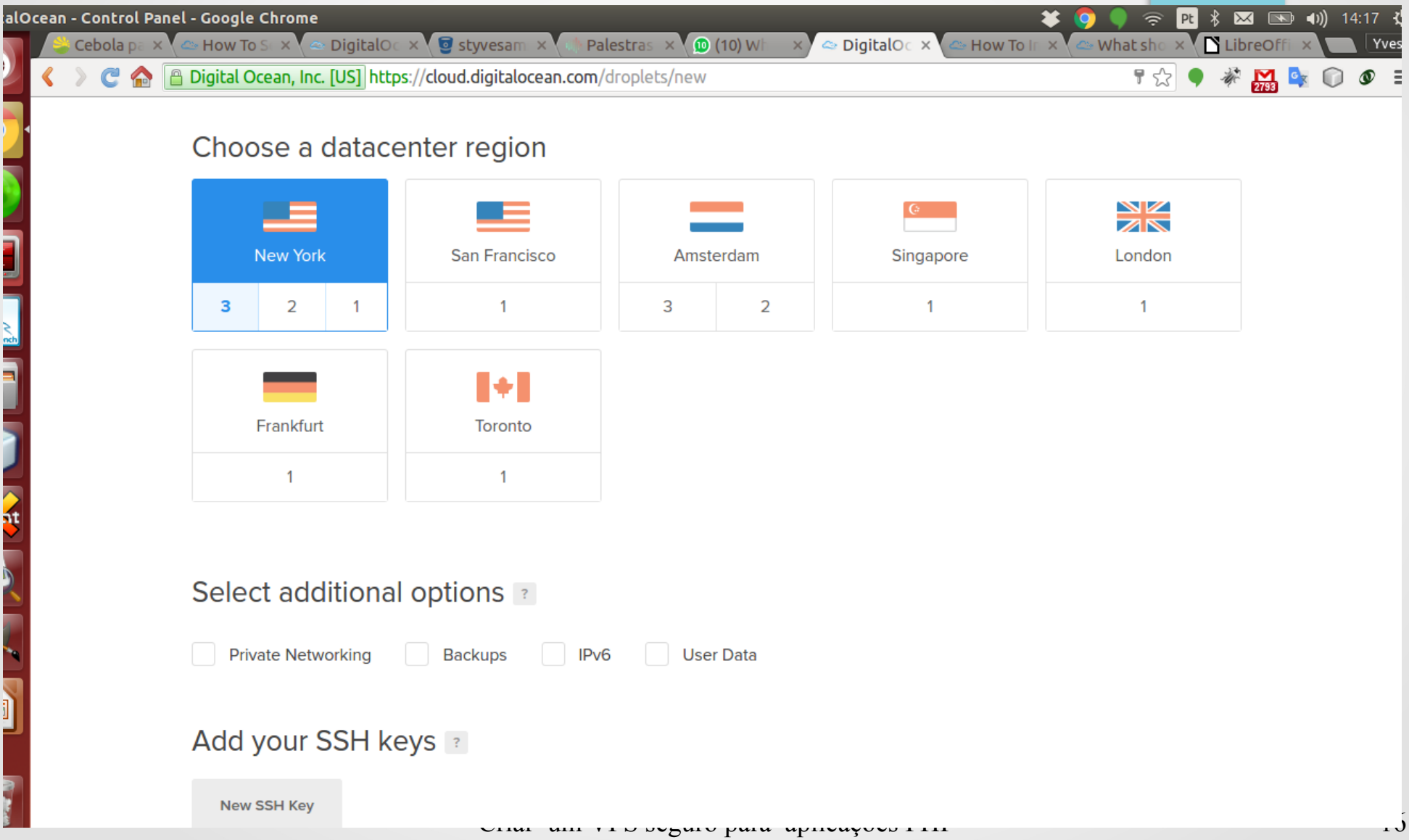
Criação de um servidor



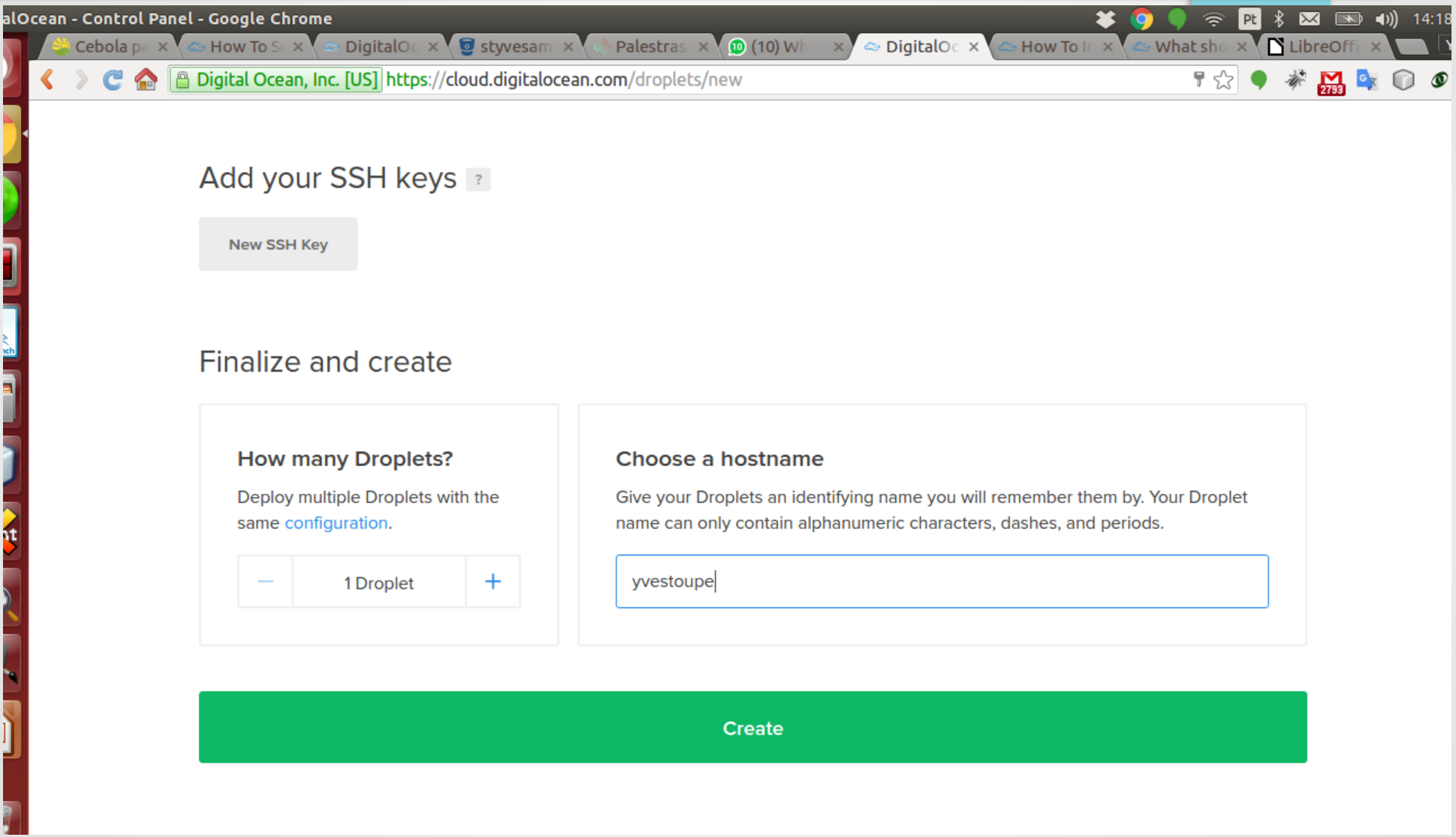
Criação de um servidor



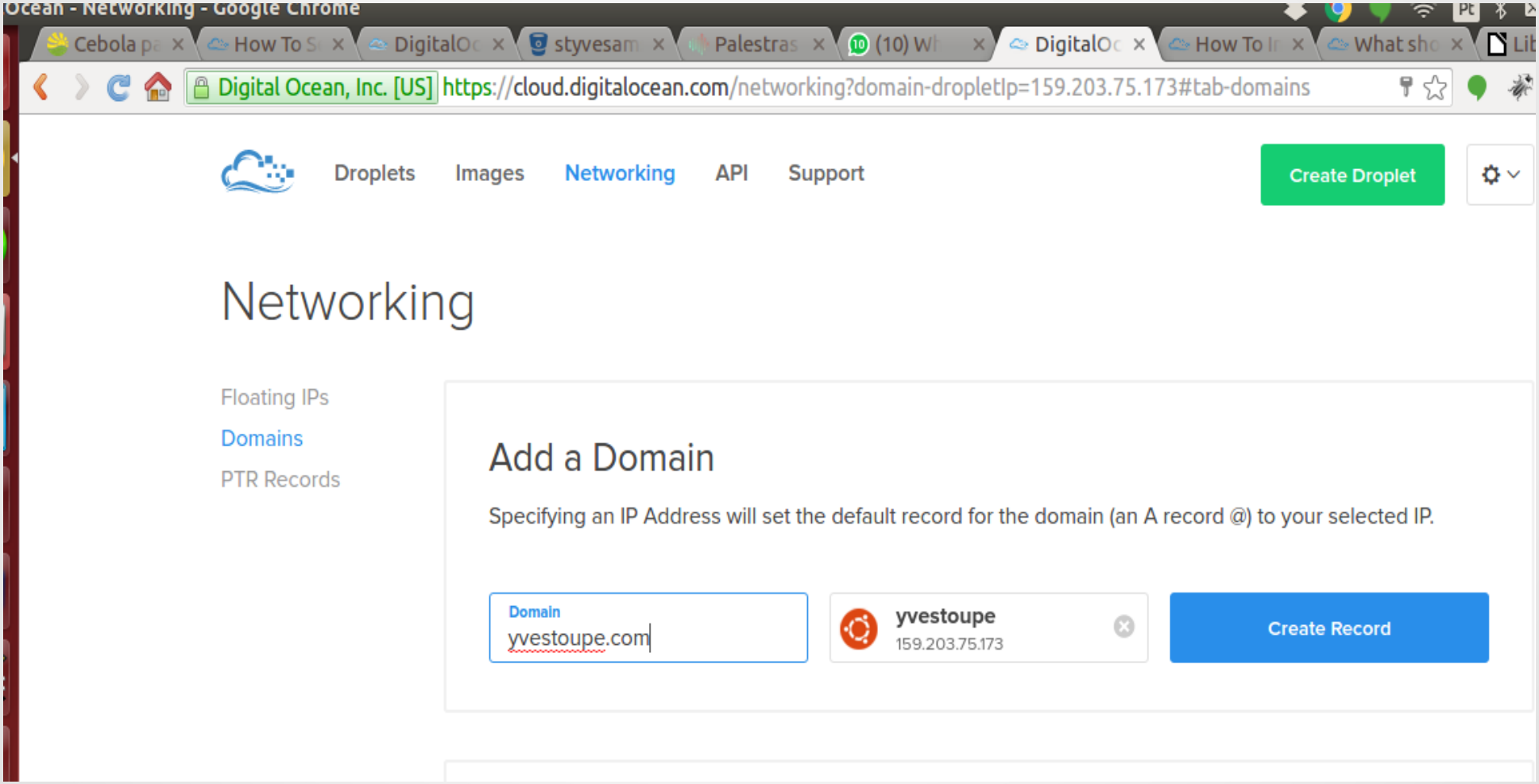
Criação de um servidor



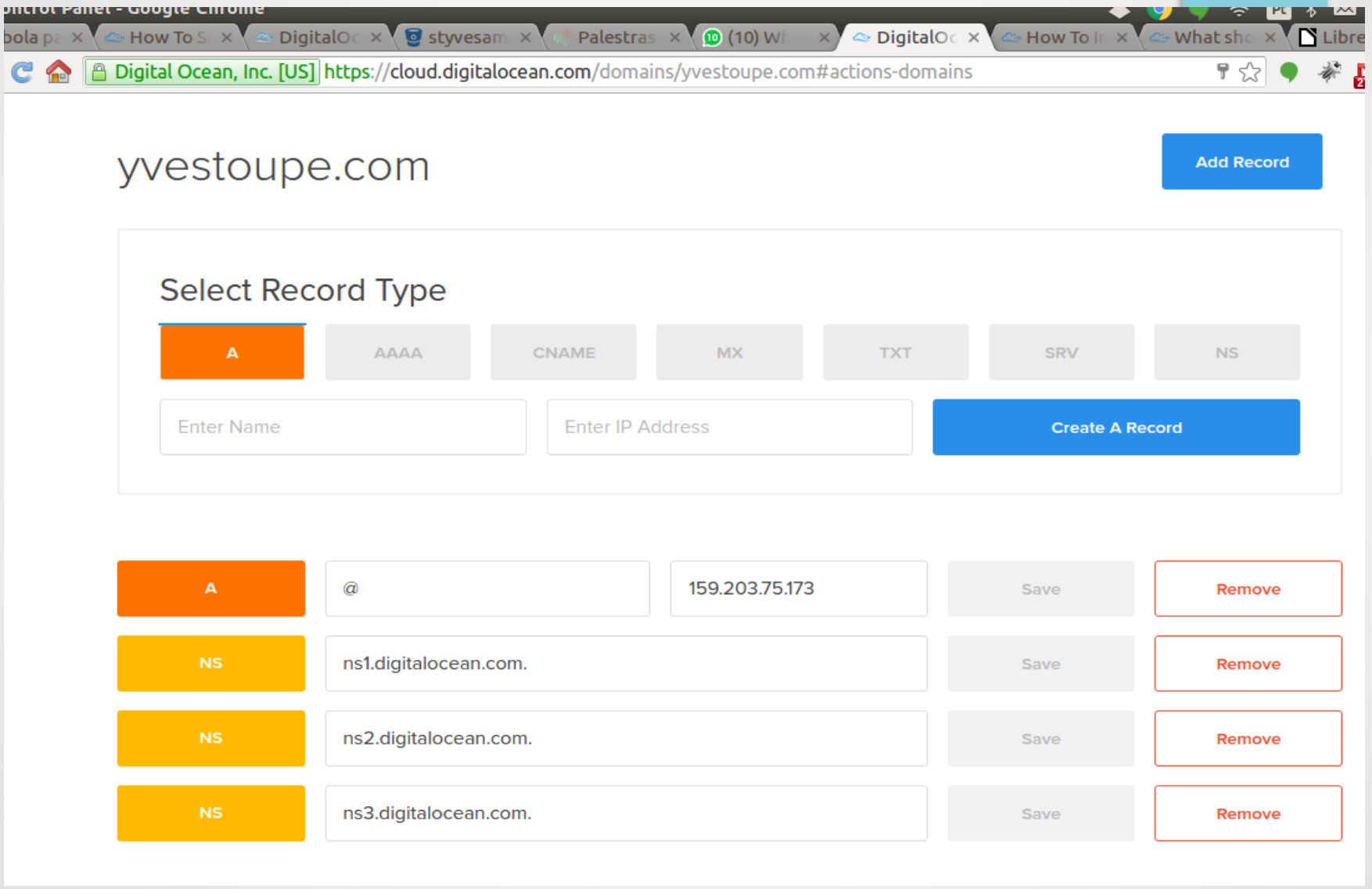
Criação de um servidor



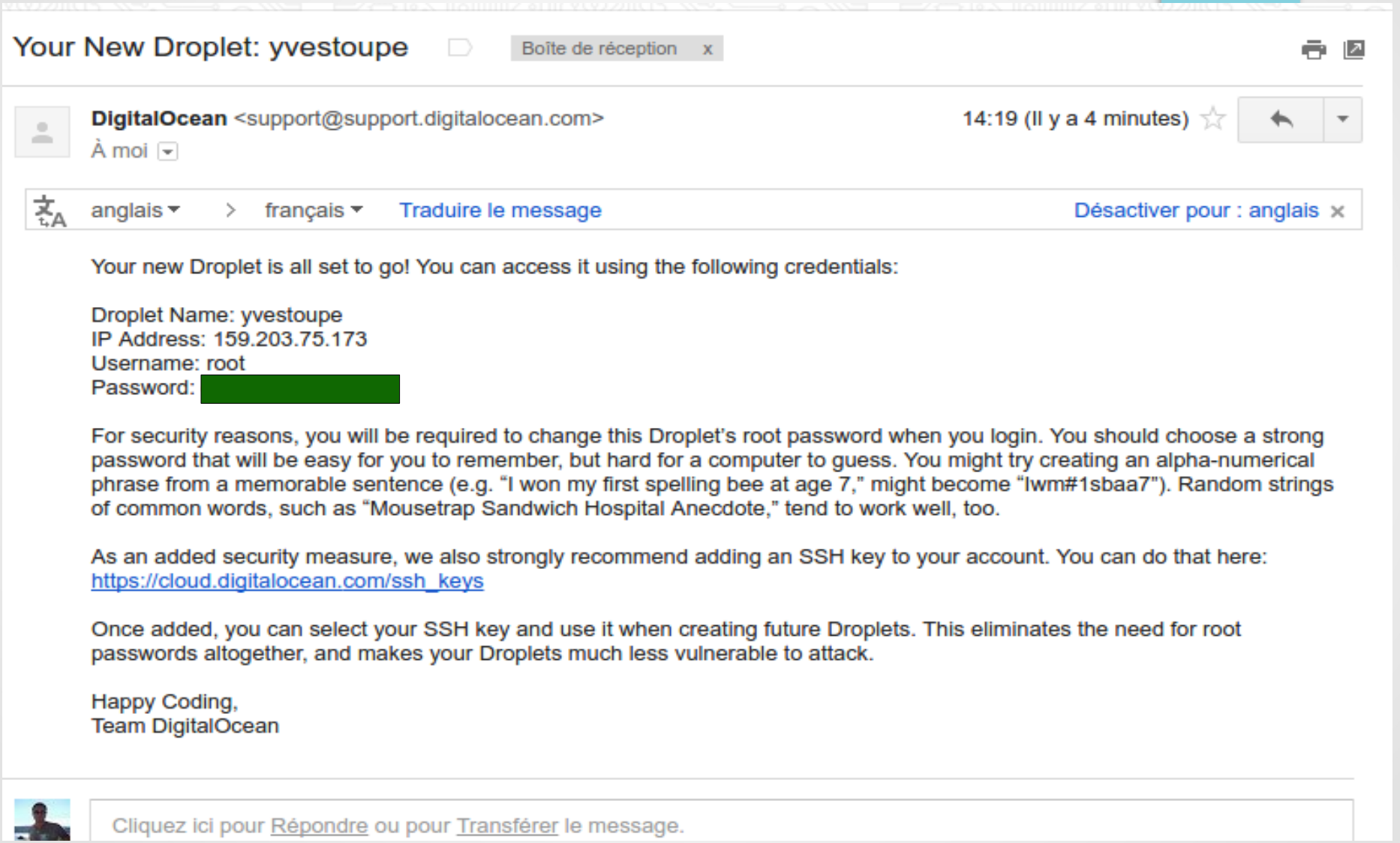
Criação de um servidor



Criação de um servidor



Criação de um servidor



Your New Droplet: yvestoupe

Boîte de réception x

DigitalOcean <support@support.digitalocean.com>

14:19 (Il y a 4 minutes)

À moi

anglais > français Traduire le message Désactiver pour : anglais

Your new Droplet is all set to go! You can access it using the following credentials:

Droplet Name: yvestoupe
IP Address: 159.203.75.173
Username: root
Password: [redacted]

For security reasons, you will be required to change this Droplet's root password when you login. You should choose a strong password that will be easy for you to remember, but hard for a computer to guess. You might try creating an alpha-numerical phrase from a memorable sentence (e.g. "I won my first spelling bee at age 7," might become "Iwm#1sbaa7"). Random strings of common words, such as "Mousetrap Sandwich Hospital Anecdote," tend to work well, too.

As an added security measure, we also strongly recommend adding an SSH key to your account. You can do that here: https://cloud.digitalocean.com/ssh_keys

Once added, you can select your SSH key and use it when creating future Droplets. This eliminates the need for root passwords altogether, and makes your Droplets much less vulnerable to attack.

Happy Coding,
Team DigitalOcean

Cliquez ici pour Répondre ou pour Transférer le message.

Se conectar ao servidor Ubuntu com SSH

Pré-requisitos :

Para se conectar a um servidor Linux remoto via SSH, você deve ter as Informações do Servidor e credenciais de login

- Nome de usuário
- Senha e / ou SSH Key
- Endereço IP do servidor:

Se conectar ao servidor Ubuntu com SSH

Informações do Servidor e credenciais de login

- Nome de usuário: O usuário padrão do administrador ou de superusuário, na maioria dos servidores Linux é root
- Senha e / ou SSH Key: A senha que é usado para autenticar o usuário. Se você adicionou uma chave SSH pública , você deve ter a chave SSH privada do par de chaves (e senha da chave , se ele tiver um)
- Endereço IP do servidor: Este é o endereço que identifica exclusivamente o seu servidor na Internet

Se conectar ao servidor Ubuntu com SSH

SSH Client Software

Há uma variedade de clientes SSH que você pode usar para se conectar a um servidor Linux. Nós vamos falar das duas seguintes:

- OpenSSH (Linux e Mac OS X): Uma coleção de software que vem com a maioria dos sistemas operacionais Unix-like, que inclui o comando ssh
- PuTTY (Windows): cliente SSH gratuito que pode ser executado no Windows, e está disponível para download na página de download PuTTY. putty.exe é o cliente SSH, e putty.exe também devem ser transferidas se você quiser usar chaves SSH.

SSH login como Root

ssh root@SERVER_IP_ADDRESS

Por exemplo, se o endereço IP do servidor foi 123.234.123.234, o comando ficaria assim:

$ ssh root@123.234.123.234.

SSH login como Root

A primeira vez que você tentar se conectar ao seu servidor, você provavelmente vai ver um aviso de que se parece com isso:

```
The authenticity of host '123.123.123.123 (123.123.123.123)'
can't be established.
ECDSA key fingerprint is
79:95:46:1a:ab:37:11:8e:86:54:36:38:bb:3c:fa:c0.
Are you sure you want to continue connecting (yes/no)?
```

Autenticação

A etapa de autenticação envolve o fornecimento de uma senha e / ou uma chave SSH privada para provar que você está autorizado a efetuar o login .

```
The authenticity of host '123.123.123.123 (123.123.123.123)' can't be established.
ECDSA key fingerprint is
79:95:46:1a:ab:37:11:8e:86:54:36:38:bb:3c:fa:c0.
Are you sure you want to continue connecting (yes/no)? Yes

yves@yves-Samsung-pc:~$ ssh root@159.203.75.173
root@159.203.75.173's password: xxxxxxx
```

Autenticação

Se você adicionou uma chave na criaçao e você tem a chave privada instalada no seu computador, OpenSSH tentará usar a chave para autenticar a conta root.

Se você usou uma chave com uma senha, você precisará fornecer a senha para concluir o processo de login.

Autenticação

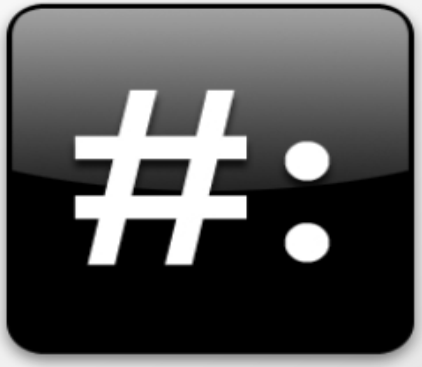
```
yves@yves-Samsung-pc:~$ ssh root@159.203.75.173
root@159.203.75.173's password:xxxxxxx
You are required to change your password immediately (root enforced)
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 3.13.0-79-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sat Mar 12 12:19:05 EST 2016

  System load: 0.0                Memory usage: 9%   Processes:       52
  Usage of /:  7.4% of 19.56GB   Swap usage:   0%   Users logged in: 0

  Graph this data and manage this system at:
    https://landscape.canonical.com/

Changing password for root.
(current) UNIX password:
```

Sobre Root

O usuário root é o usuário administrativo em um ambiente Linux que tem privilégios muito amplos.

Por causa dos privilégios elevado da conta root, nao é recomendado seu uso no dia dia .

Isso ocorre porque parte do poder da conta root é a capacidade de fazer mudanças muito destrutivas, mesmo por acidente.



Passo dois - Criar um novo usuário

Será solicitado algumas perguntas, começando com a senha da conta.

Digite uma senha forte e, opcionalmente, preencher qualquer informação adicional. Isso não é necessário e você pode apenas clicar em "ENTER" em qualquer campo que deseja ignorar.

```
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for yves
Enter the new value, or press ENTER for the default
        Full Name []: darkmra 2016
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

Criar um novo usuário

O próximo passo é a criação de uma conta de usuário alternativa com um escopo reduzido de influência para o trabalho do dia-a-dia.

Vamos ensiná-lo a dar os privilégios necessarios quando precisar

Passo dois - Criar um novo usuário

Este exemplo cria um novo usuário chamado "darkmira", mas você deve substituí-lo por um nome de usuário que você gosta

```
root@yvestoupe:~# adduser darkmira

Enter the new value, or press ENTER for the default
    Full Name []: darkmra 2016
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

Passo dois - Criar um novo usuário

Será solicitado algumas perguntas, começando com a senha da conta.

Digite uma senha forte e, opcionalmente, preencher qualquer informação adicional. Isso não é necessário e você pode apenas clicar em "ENTER" em qualquer campo que deseja ignorar.

```
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for yves
Enter the new value, or press ENTER for the default
        Full Name []: darkmra 2016
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

Passo três - privilégios de root

Agora, temos uma nova conta de usuário com privilégios de conta regulares. Mas, podemos às vezes precisar fazer tarefas administrativas.

Para evitar ter que sair do nosso usuário normal e voltar a iniciar sessão como a conta root, podemos definir o que é conhecido como "super utilizador" ou privilégios de root para a nossa conta normal. Isto permitirá que o nosso usuário normal consegue executar comandos com privilégios administrativos, colocando a palavra sudo antes de cada comando.

Passo três - privilégios de root

Para adicionar esses privilégios ao nosso novo usuário, precisamos adicionar o novo usuário ao grupo "sudo".

Por padrão, no Ubuntu, os usuários que pertencem ao grupo "sudo" estão autorizados a utilizar o comando sudo.

```
# gpasswd -a darkmira sudo
Adding user darkmira to group
sudo
```

Agora o usuário pode executar comandos com privilégios de superusuário !!!

Passo 4 - Adicionar chave pública

(Recomendado)

O próximo passo na segurança de seu servidor é para configurar a autenticação de chave pública para o seu novo usuário.

Essa configuração irá aumentar a segurança do seu servidor, exigindo uma chave SSH privada para fazer o login.

Gerar um par de chaves

Para gerar um novo par de chaves, digite o seguinte comando no terminal de sua máquina local (ou seja, seu computador.):

```
localuser$  ssh-keygen
```

Assumindo que o seu utilizador local é chamado de "localuser", você verá uma saída que se parece com o seguinte:

```
ssh-keygen output
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/localuser/.ssh/id_rsa):
```

Copiar a chave pública para o servidor

Usando ssh-copy-id

Se a sua máquina local tem o script ssh-copy-id instalado, você pode usá-lo para instalar sua chave pública para qualquer usuário que tem os credenciais de login .

Execute o script ssh-copy-id, especificando o endereço de usuário e IP do servidor que você deseja instalar a chave no, como este:

```
$      ssh-copy-id darkmira @SERVER_IP_ADDRESS
```

Passo 5 - Configurar SSH Daemon

Modificando sua configuração SSH daemon para impedir o acesso SSH remoto para a conta root.

```
nano /etc/ssh/sshd_config
```

Modificar essa linha para desabilitar o login do root:

```
PermitRootLogin no
```

Passo 5 - Configurar SSH Daemon

A desativação login remoto do login da conta root é altamente recomendado em todos os servidores!

Passo Seis - Recarregar SSH

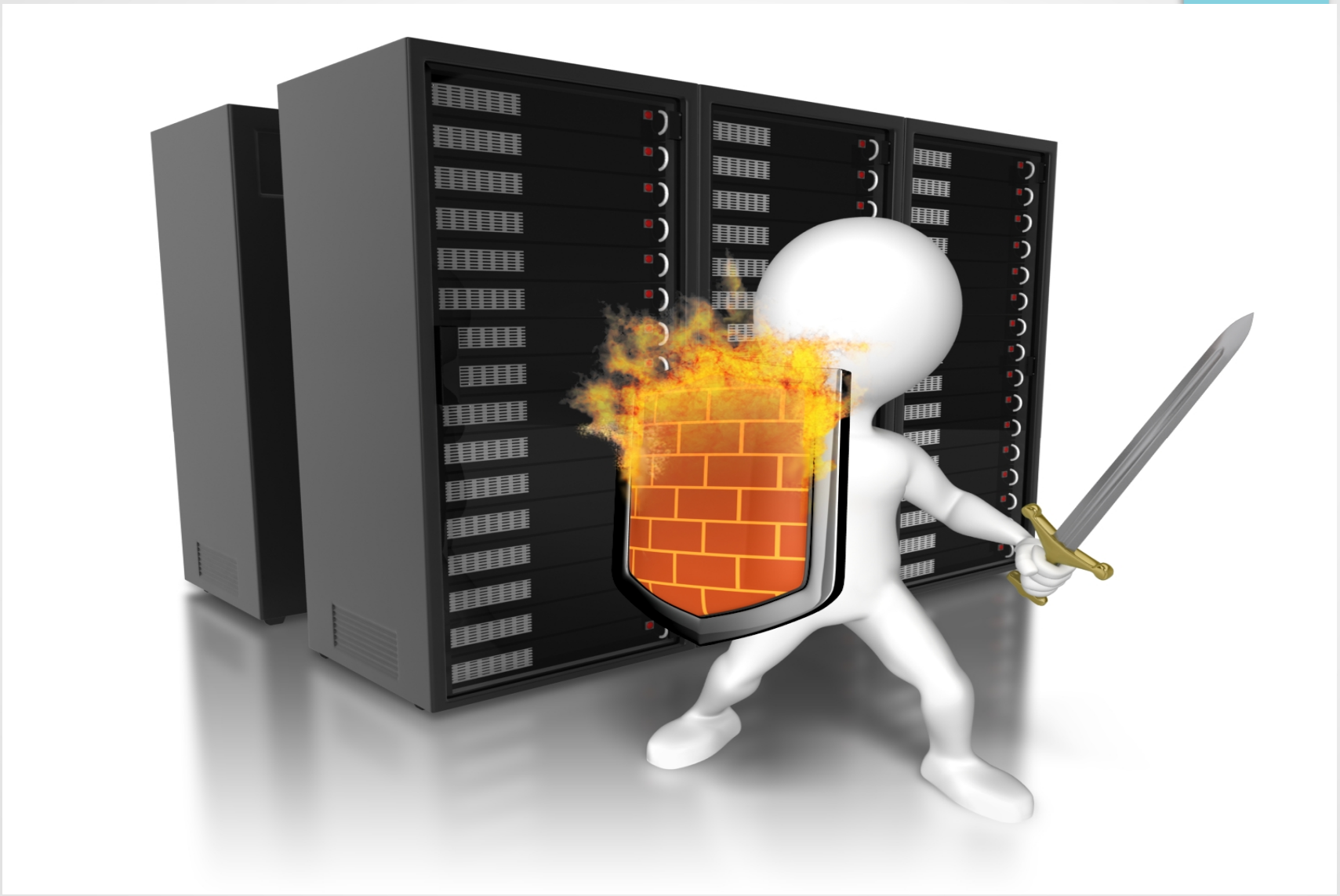
Agora que fizemos a nossa mudança, é preciso reiniciar o serviço SSH para usar a nossa nova configuração.

Digite isso para reiniciar SSH:

```
$ service ssh restart
```

Agora o usuário pode executar comandos com privilégios de superusuário !!!

Passos adicionais para novo Servidor Ubuntu



Configurando um firewall básico

Firewalls fornecer um nível básico de segurança para o servidor. Estas aplicações são responsáveis por negar o tráfego para cada porta no seu servidor com exceções das portas / serviços que você aprovadou.

O Ubuntu vem com uma ferramenta chamada ufw que pode usar para configurar as políticas de firewall.

Configurando um firewall básico

Nossa estratégia básica será de bloquear tudo o que não temos uma boa razão para manter aberta.

Em primeiro lugar, precisamos criar uma exceção para conexões SSH para que possamos manter o acesso para administração remota.

Configurando um firewall básico

O daemon SSH é executado na porta 22 por padrão e ufw pode implementar uma regra pelo nome, se o padrão não foi alterado. Vamos habilitar a exceção, digitando:

```
local$ sudo ufw allow ssh
```

Configurando um firewall básico

Esta é a configuração de firewall mínimo.

Ele só vai permitir o tráfego na porta de SSH e todos os outros serviços estarão inacessíveis.

Se você planeja executar serviços adicionais, você vai precisar para abrir o firewall em cada porta necessária.

Configurando um firewall básico

Se você planeja executar um servidor HTTP web convencional, você vai precisar para permitir o acesso à porta 80:

```
local$ sudo ufw allow 80/tcp
```

Configurando um firewall básico

Se você pretende executar um servidor web com SSL / TLS habilitado, você deve permitir o tráfego para essa porta também

```
local$  sudo ufw allow 443/tcp
```

Configurando um firewall básico

Se você precisa de e-mail SMTP , a porta 25 deverá ser aberto:

```
local$  sudo ufw allow 25/tcp
```

Configurando um firewall básico

Depois que você terminar de adicionar as exceções, você pode rever suas seleções, digitando:

```
local$  sudo ufw show added
```

Se tudo estiver em ordem, você pode ativar o firewall, digitando:

```
local$  sudo ufw enable
```

Configurar Time Zones e NTP

O primeiro passo será garantir que seu servidor está operando sob o fuso horário correto.

O segundo passo será configurar o sistema para sincronizar seu relógio para o tempo padrão mantido por uma rede global de servidores NTP. Isto ajudará a prevenir alguns comportamentos inconsistentes que podem surgir a partir de relógios fora de sincronia.

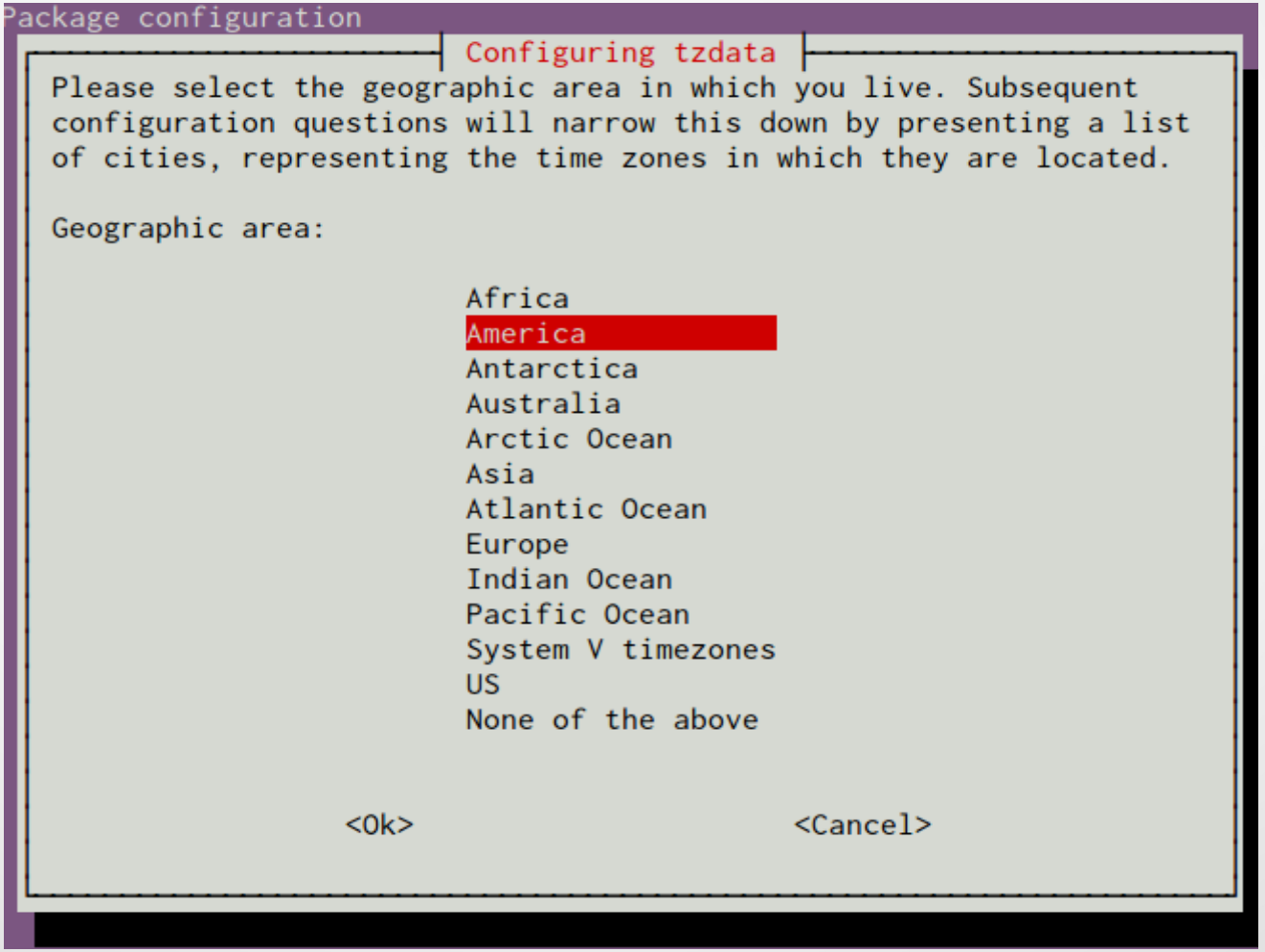
Configurar Time Zones

Nosso primeiro passo é definir fuso horário do nosso servidor. Este é um procedimento muito simples que pode ser realizado configurando o pacote tzdata:

```
local$  sudo dpkg-reconfigure tzdata
```

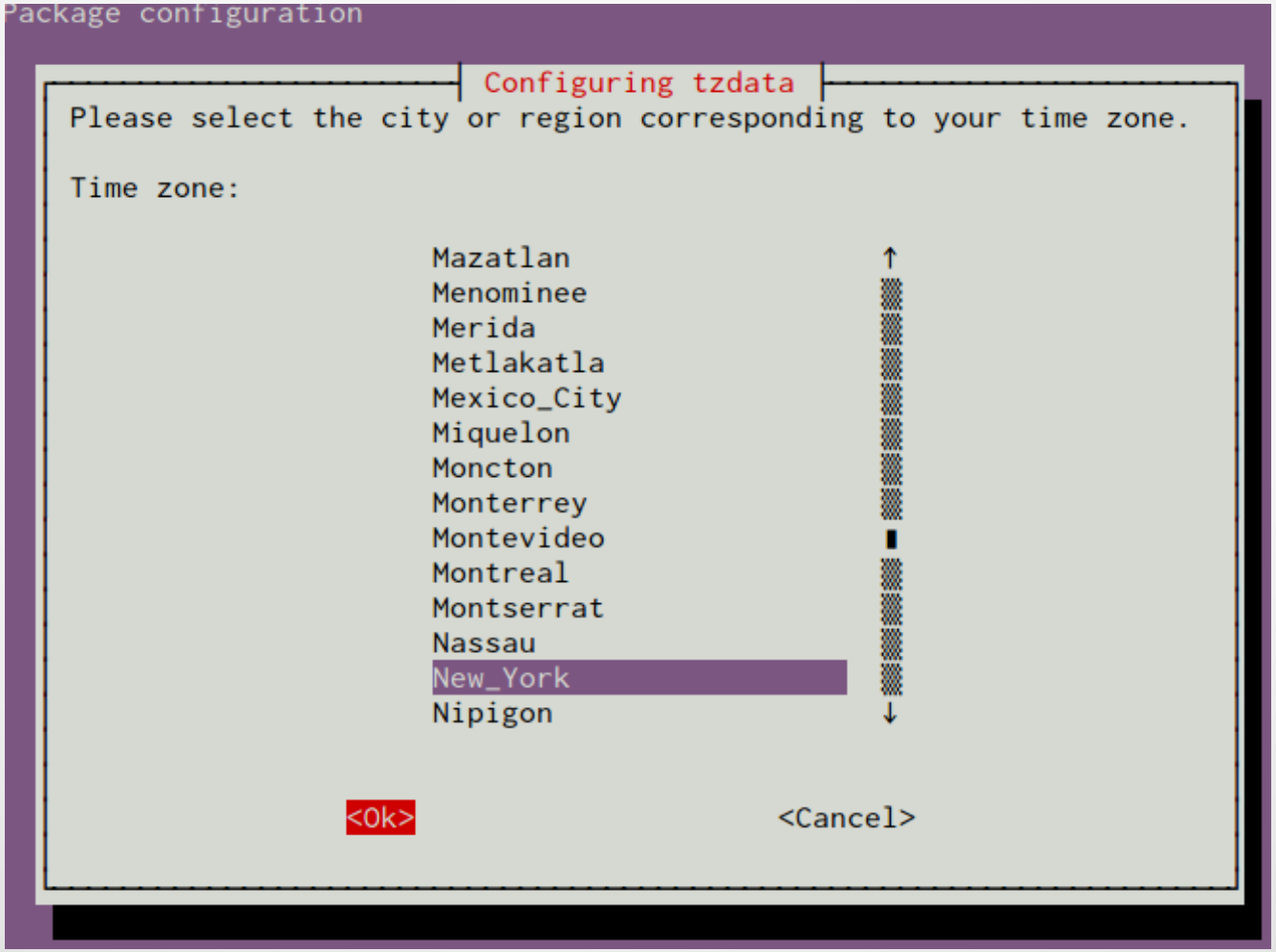
Configurar Time Zones

Será apresentado com um sistema de menu que permite que você selecione a região geográfica do seu servidor:



Configurar Time Zones

Depois de selecionar uma área, você terá que escolher o fuso horário específico que é apropriado para seu servidor:



Configurar Time Zones

Seu sistema será atualizado para usar o fuso horário selecionado, e os resultados serão impressos na tela:

```
Current default time zone: 'America/Sao_Paulo'
Local time is now:      Mon  Mar  3 17:00:11 EST 2016.
Universal Time is now:  Mon Mar  3 22:00:11 UTC 2016.
```

Em seguida, vamos configurar NTP.

Configurar NTP sincronização

A configuração de NTP permitirá que seu servidor fica em sincronia com outros servidores, permitindo muitas operações que dependem de ter o tempo correto.

Para a sincronização NTP, vamos usar um serviço chamado NTP, que pode instalar a partir de repositórios padrão do Ubuntu:

```
sudo apt-get update
sudo apt-get install ntp
```

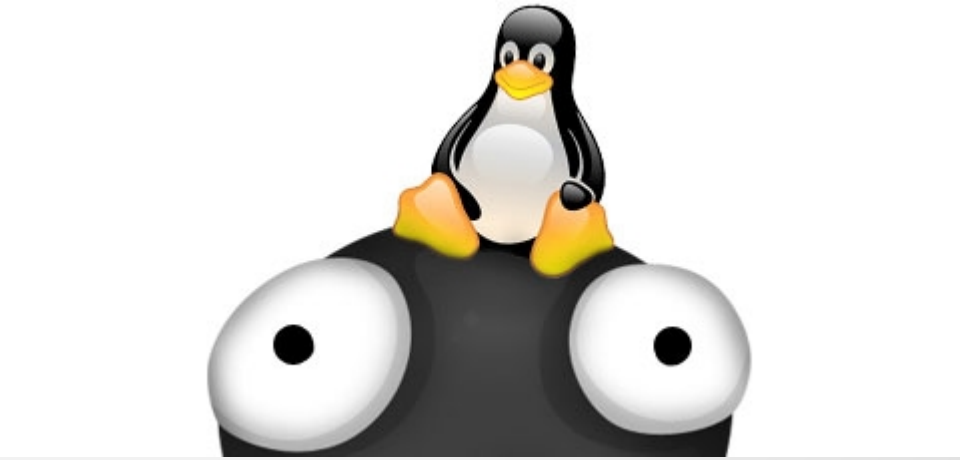
Configurar NTP sincronização

Isto é tudo que você tem que fazer para configurar a sincronização NTP no Ubuntu.

O daemon iniciará automaticamente a cada inicialização e continuamente ajustar a hora do sistema para estar em linha com os servidores NTP globais ao longo do dia.

Onde ir a partir daqui?

Neste ponto, Você pode instalar qualquer software que você precisa no seu servidor agora.



Onde ir a partir daqui?

Substitua seu próprio nome de usuário e endereço IP do servidor para poder se conectar com ssh ao servidor

```
Local$ ssh darkmira@SERVER_IP_ADDRESS
```

Onde ir a partir daqui?

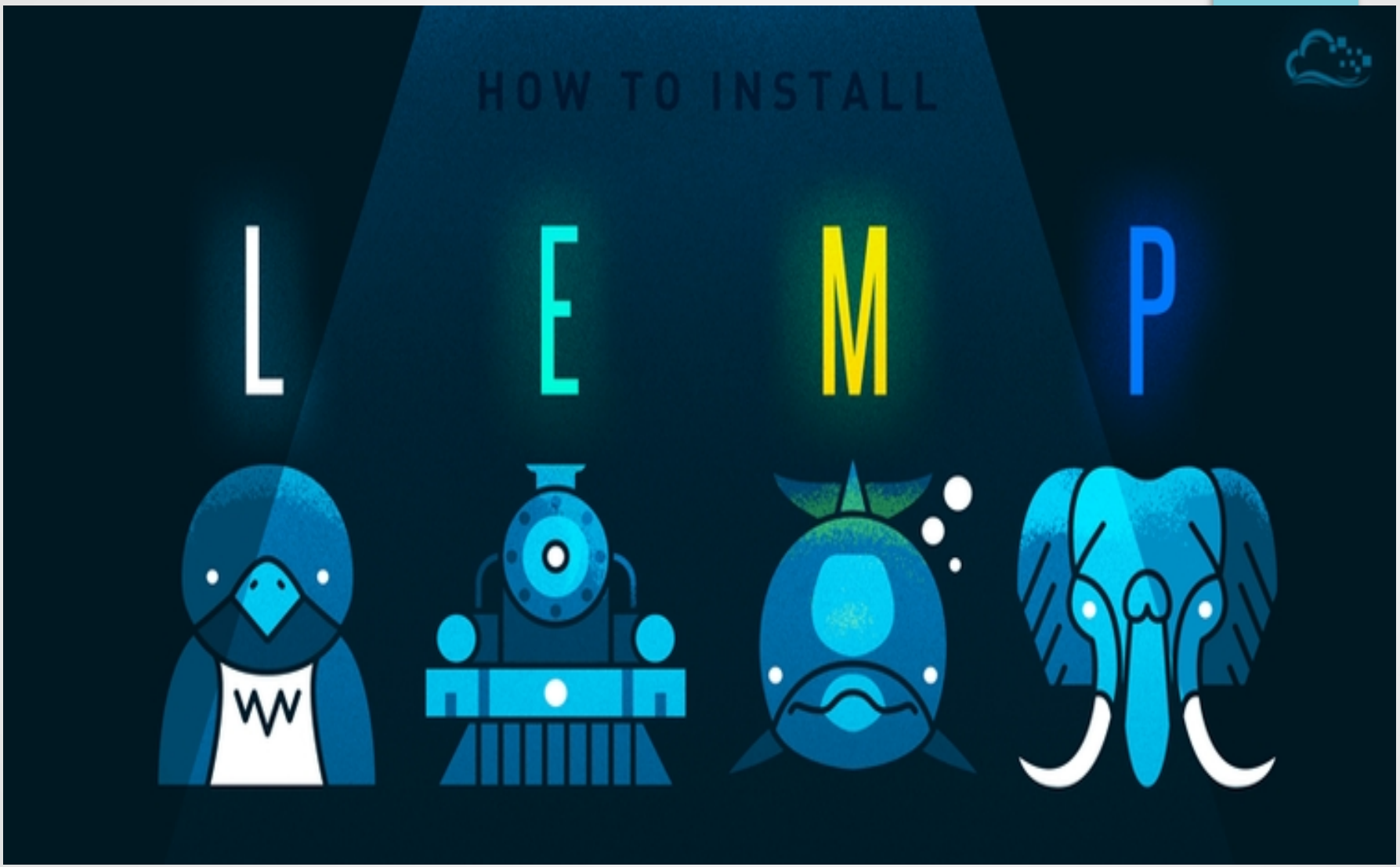
Lembre-se, se você precisa executar um comando com privilégios de root, digite "sudo" antes assim:

```
$     sudo command_to_run
```

Para sair de suas sessões, basata digitar:

```
$     exit
```

Como instalar LEMP no Ubuntu



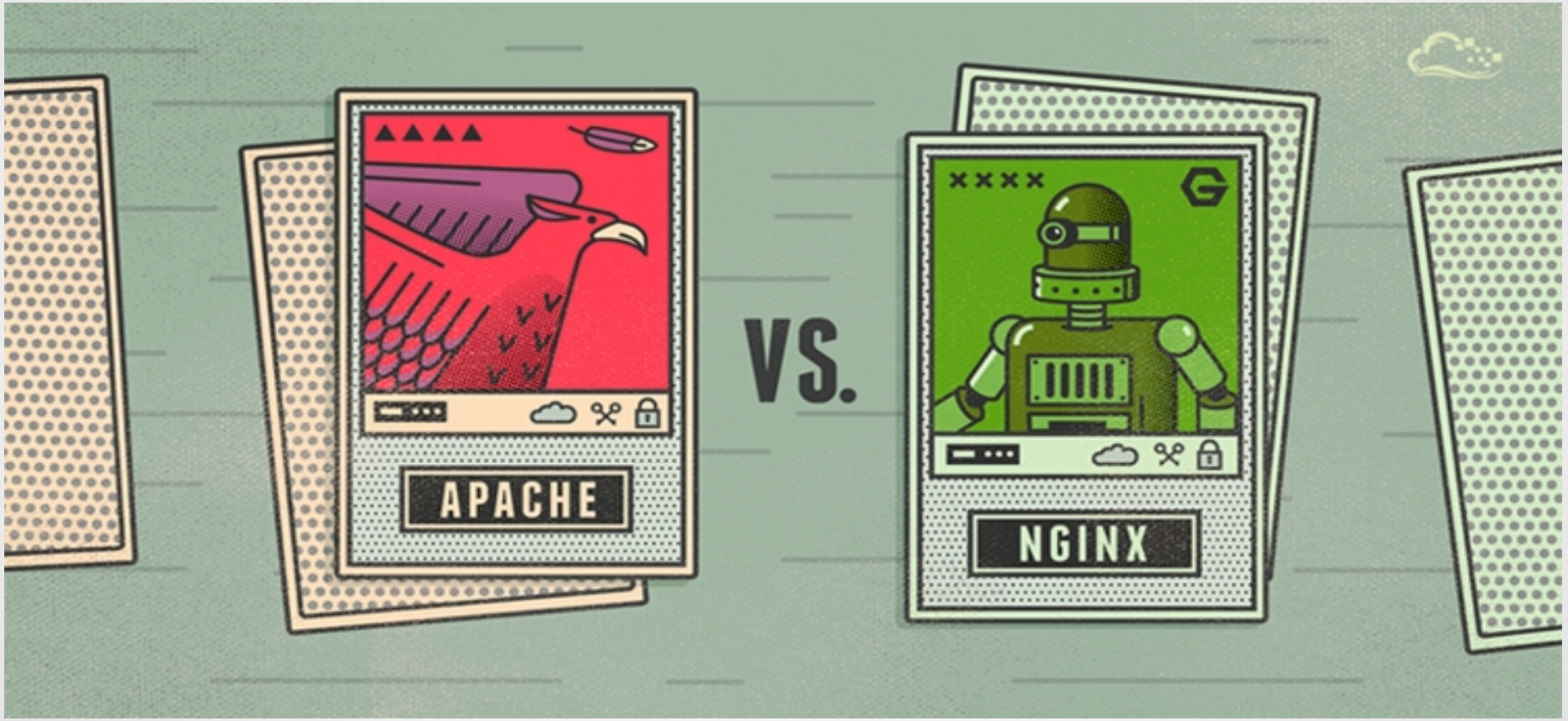
Como instalar LEMP no Ubuntu

Introdução

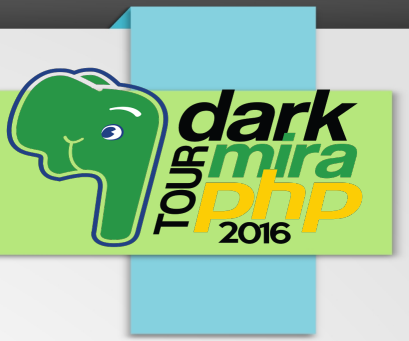
LEMP: Esta sigla descreve um sistema operacional Linux, com um servidor web Nginx. Os dados de back-end sao armazenado no MySQL e o processamento dinâmico é tratado pelo PHP.

vamos ver como instalar uma pilha toda em um servidor Ubuntu 14.04.

Apache vs Nginx



Apache vs Nginx



Apache e Nginx são os dois servidores web open source mais comuns no mundo. Juntos, eles são responsáveis por servir mais de 50% do tráfego na internet. Ambas as soluções são capazes de lidar com diversas cargas de trabalho e agir com outro software para fornecer uma pilha web completa.

Embora Apache e Nginx compartilham muitas qualidades.

É importante compreender como cada um funciona antes de escolher seu servidor web de acordo com as necessidades seus projeto .

Passo 1- Instale o Servidor Web Nginx

Sempre começar atualizando seu índice de pacotes local

```
$ sudo apt-get update
```

```
$ sudo apt-get install nginx
```

Passo 1- Instale o Servidor Web Nginx

No Ubuntu 14.04, Nginx está configurado para começar a rodar após a instalação.

Você pode testar se o servidor está instalado e funcionando, acessando nome de domínio do seu servidor ou o endereço IP público de seu navegador web.

Passo 1- Instale o Servidor Web Nginx

Se você não tem um nome de domínio apontado para o servidor e você não sabe o endereço IP público do seu servidor, você pode encontrá-lo digitando ...

```
ip addr show eth0 | grep inet | awk '{ print $2; }' | sed 's/\/.*$//'
```

```
111.111.111.111
fe80::601:17ff:fe61:9801
```

Passo 1- Instale o Servidor Web Nginx

http://server_domain_name_or_IP

Passo 2 - Instalar MySQL

Agora que temos um servidor web, e precisamos instalar o MySQL, um sistema de gerenciamento de banco de dados, para armazenar e gerenciar os dados para o nosso site.

```
$    sudo apt-get install mysql-server
```

Passo 2 - Instalar MySQL

MySQL está instalado, mas a sua configuração não é exatamente completa ainda.

Passo 2 - Instalar MySQL

Em primeiro lugar,
precisamos dizer ao MySQL de gerar a estrutura de diretórios que ele precisa para armazenar seus bancos de dados e informações. Nós podemos fazer isso digitando:

```
$    sudo mysql_install_db
```

Passo 2 - Instalar MySQL

Em seguida,
você vai executar esse script simples de segurança que irá pedir-lhe para modificar alguns padrões inseguros de MySQL.

```
$   sudo mysql_secure_installation
```

Você terá que inserir a senha root do MySQL que você selecionou durante a instalação.

Passo 3 - Instalar PHP

Nginx não contém processador de PHP nativa como alguns outros servidores web, vamos precisar instalar php7.0-fpm, que significa "gerenciador de processos fastCGI".

```
$  sudo apt-get install php7.0-fpm php7.0-mysql
```

Podemos instalar este módulo adicional que permitirá a PHP se comunicar com o nosso banco de dados.

Configurar o processador PHP

Precisamos fazer uma pequena alteração na configuração para tornar seu servidor seguro

Abra o arquivo de configuração principal php7.0-fpm com privilégios de root:

```
sudo nano /etc/php7.0pm/php.ini
```

Configurar o processador PHP

O que estamos procurando neste arquivo é o parâmetro que define cgi.fix_pathinfo. Este será comentada com um ponto e vírgula (;) e definido como "1" por padrão.

```
;cgi.fix_pathinfo=1
```

Configurar o processador PHP

Este é um ajuste extremamente insegura porque deixa PHP executar o arquivo mais próximo que ele pode descobrir se um arquivo PHP não corresponder exatamente.

Isso basicamente permitiria que os usuários criam solicitações PHP de uma forma que lhes permita executar scripts que eles não devem ser autorizados a executar.

Configurar o processador PHP

Vamos mudar ambas a condição e descomentar a linha

```
cgi.fix_pathinfo= 0
```

Salvar e fechar o arquivo quando terminar.

Configurar o processador PHP

Agora, só precisamos reiniciar o processador PHP digitando:

```
sudo service php7.0-fpm restart
```

Passo 4 - Configurar Nginx para usar nosso processador PHP

Agora precisamos dizer a Nginx de usar nosso processador PHP para conteúdo dinâmico.

Passo 4 - Configurar Nginx para usar nosso processador PHP

Vamos abrir o arquivo de configuração padrão Nginx servidor bloco digitando:

```
sudo nano /etc/nginx/sites-available/default
```

Passo 4 - Configurar Nginx para usar nosso processador PHP

Atualmente, com os comentários removidos, o arquivo de bloco de servidor padrão Nginx parece com isso:

```
server {
    listen 80 default_server;
    listen [::]:80 default_server ipv6only=on;

    root /usr/share/nginx/html;
    index index.html index.htm;

    server_name localhost;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Passo 4 - Configurar Nginx para usar nosso processador PHP

Precisamos fazer algumas alterações neste arquivo

- Em primeiro lugar, precisamos adicionar uma opção index.php como o primeiro valor da nossa directiva índice para permitir que o arquivos de index.php seja servido quando um diretório é solicitada.

Passo 4 - Configurar Nginx para usar nosso processador PHP

- Precisamos modificar a directiva server_name para apontar para nome de domínio do nosso servidor ou endereço IP público.

Passo 4 - Configurar Nginx para usar nosso processador PHP

- O arquivo de configuração atual inclui algumas linhas comentadas que definem as rotinas de processamento de erro. Vamos incluir essa funcionalidade.

Passo 4 - Configurar Nginx para usar nosso processador PHP

- Para o processamento real de PHP , vamos incluir outra seção.

Vamos adicionar uma diretiva try_files para se certificar de que Nginx não passa solicitações erradas para o nosso processador PHP.

```
server {
    listen 80 default_server;
    listen [::]:80 default_server ipv6only=on;

    root /usr/share/nginx/html;
    index index.php index.html index.htm;

    server_name server_domain_name_or_IP;

    location / {
        try_files $uri $uri/ =404;
    }

    error_page 404 /404.html;
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root /usr/share/nginx/html;
    }

    location ~ \.php$ {
        try_files $uri =404;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

Passo 5 - Criar um arquivo PHP para configuração de teste

Precisamos testar que Nginx pode corretamente rodar arquivos .php para o nosso processador PHP.

Podemos fazer isso criando um arquivo PHP de teste na raiz do documento.

```
$ sudo nano /usr/share/nginx/html/info.php
```

Passo 5 - Criar um arquivo PHP para configuração de teste

Pode ser um phpinfo() para retornar informação sobre o nosso servidor:

```
<?php
phpinfo();
?>
```

Passo 5 - Criar um arquivo PHP para configuração de teste

pode visitar esta página no seu navegador web, visitando com o nome de domínio do seu servidor ou o endereço IP público seguido por /info.php:

```
http://server_domain_name_or_IP/info.php
```

Passo 5 - Criar um arquivo PHP para configuração de teste

Se você vê uma página que se parece com isso, você configurou o processamento PHP com Nginx com sucesso.

PHP Version 5.5.9-1ubuntu4

System	Linux lemp 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 x86_64
Build Date	Apr 9 2014 17:10:12
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/fpm
Loaded Configuration File	/etc/php5/fpm/php.ini
Scan this dir for additional .ini files	/etc/php5/fpm/conf.d
Additional .ini files parsed	/etc/php5/fpm/conf.d/05-opcache.ini, /etc/php5/fpm/conf.d/10-pdo.ini, /etc/php5/fpm/conf.d/20-json.ini, /etc/php5/fpm/conf.d/20-mysql.ini, /etc/php5/fpm/conf.d/20-mysqli.ini, /etc/php5/fpm/conf.d/20-pdo_mysql.ini
PHP API	20121113
PHP Extension	20121212
Zend Extension	220121212
Zend Extension Build	API220121212,NTS

Passo 5 - Criar um arquivo PHP para configuração de teste

Depois de testar isso, provavelmente é melhor para remover o arquivo que você criou, pois ele pode realmente dar usuários não autorizados algumas dicas sobre a configuração que podem ajudá-los a tentar invadir seu servidor .
Você sempre pode regenerar este arquivo se você precisar dele mais tarde.

remover o arquivo digitando:

```
$  sudo rm /usr/share/nginx/html/info.php
```

Conclusão

Agora você viu como criar um LEMP configurado no seu servidor Ubuntu 14.04 seguro .
Isso lhe dá uma base muito flexível para servir conteúdo web para os seus visitantes.

Obrigado …

Criando um VPS seguro para suas aplicações PHP