



Professor Leandro <lseverino@gmail.com>

dados algodao

1 message

Professor Leandro <lseverino@gmail.com>
To: Leandro Severino <lseverino@gmail.com>

Sat, Feb 9, 2019 at 7:54 AM

1) Depois do e-mail de boas vindas:

E-mail: para receber os contatos do form: escolainfantilalgodaodocesite@gmail.coma) Solicita ao cliente criar a conta no registro.br
e também na VPS. E pagar para ativar.

b) Depois ele nos passa os dados de login neste dois serviços.

[Acesso a hospedagem]

1) Painel Administrativo:

URL: <http://solus.bhservers.com.br:5353>

Usuário: bh3899

Senha: Luana1015

<https://cliente.bhservers.com.br/>user: escolainfantilalgodaodoce@hotmail.com

pwd: luana1015

[Acesso ao registro.br]

usr: PASAL257

pwd: Luana1015

<https://registro.br/2/login>algodaodocecanoas.com.br A 167.114.133.128www.algodaodocecanoas.com.br CNAME algodaodocecanoas.com.br

[Configuração do Servidor VPS na BhServers]

(OK) a) Atualizar o linux (a versão do Ubuntu padrão para a mais recente)

(OK) 1. apt update

(OK) 2. apt upgrade

(OK) 2. apt-get install update-manager-core

(OK) 3. do-release-upgrade (16.04 no momento (jun/2017) que foi criado este tutorial)

(OK) 4. restart now

(OK) 5. repetir de 1 à 3

(OK) 6. apt-get autoremove && apt-get autoclean

(OK) a.1) Instalar o NGINX

(OK) apt install nginx

[Procedimentos para configurar o suporte ao PHP para envio de e-mail]

(OK) 1. sudo apt install mailutils

(OK) 2. sudo apt install php-fpm

(OK) 3. Descomentar no arquivo /etc/nginx/sites-available/default o suporte ao PHP

(OK) 4. Reiniciar o serviço do nginx

(OK) a.2) No registro.br

(OK) - Editar zona de DNSs e:

(OK) Criar registro A apontando para o IP público do servidor.

(OK) Criar registro www [cname] apontando para o domínio [Nome do registro A].

(OK) - Esperar uns 15 minutos e domínio com e sem www já deve estar respondendo, pode testar/validar em: <https://www.whatsmydns.net/#A/algodaodocecanoas.com.br>

(OK) a.3) Procedimentos de segurança no servidor

(OK) ssh root@167.114.133.128

(OK) usr: root

(OK) pwd: Leandro2892

(OK) ssh algodao@167.114.133.128

(OK) usr: algodao

(OK) pwd: Leandro2892

(OK) 1) Não permitir login/acesso via SSH como root

(OK) Editar o arquivo /etc/ssh/sshd_config

(OK) alterar a chave PermitRootLogin no

(OK) service ssh restart

(OK) 2) Criar usuário para operar o servidor e colocar ele no grupo SUDOUSERS

(OK) adduser algodao sudo

ou já para um usuário existente:

(OK) usermod -aG sudo algodao

(OK) 3) Criar chave SSH para automatizar o login do usuário criado.

(OK) No servidor na pasta /home/algodao/.ssh:

(OK) - Criar o arquivo authorized_keys

(OK) - Copiar o conteúdo do arquivo id_rsa.pub da minha máquina local para dentro deste arquivo, salvar e fechar ele

(OK) - chmod 600 authorized_keys

(OK) Não permitir login via ssh com senha

(OK) - Editar o arquivo /etc/ssh/sshd_config

(OK) alterar a chave PasswordAuthentication no

(OK) - service ssh restart

(OK) 4) Não permitir login de root via ssh

(OK) Editar o arquivo /etc/ssh/sshd_config

(OK) alterar a chave PermitRootLogin no

(OK) service ssh restart

[Perfumarias]

5) Corrigir bug no apt update que acontece após a atualização da versão do S.O

6) Instalar o certificado SSH let_encrypts

7) Estudar e Configurar procedimentos de segurança no NGINX.

(OK) b) Instalar o OpenJDK

(OK) add-apt-repository ppa:openjdk-r/ppa

(OK) apt-get update
(OK) apt-get install openjdk-8-jdk

[Caso existam mais java instalados]

(OK) update-alternatives --config java
(OK) update-alternatives --config javac

(OK) Configurar a variável de ambiente JAVA_HOME:
(OK) Editar o arquivo /etc/profile:
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64 (path para a instalação do JDK)

[Instalar o SDKMan]

- Ver em <http://sdkman.io/index.html>
curl -s "https://get.sdkman.io" | bash

Ob: sair e entrar novamente via ssh para carregar o bash novamente.

- Instalar o maven:
sdk install maven

Ob: Testar com mvn

c) Instalar o PostgreSQL

<https://docs.docker.com/engine/installation/linux/docker-ce/ubuntu/#install-from-a-package>

sudo apt-get install postgresql postgresql-contrib

createdb algodao

sudo -u postgres createuser --interactive

psql
GRANT ALL PRIVILEGES ON DATABASE algodao TO usr_algodao;
\q

DATABASE bd_algodao WITH OWNER usr_algodao

bd: bd_algodao
usr: usr_algodao
ou usr_bd_algodao
pwd: algodao#12131415

usr: postgres
pwd: algodao#12131415

(OK) d) Instalar o Git

(OK) e) Fazer o clone do projeto

(OK) Criar as chaves ssh privadas e publicas no servidor de hospedagem
ssh-keygen

(OK) Liberar as chaves SSHs para acesso do servidor ao repositório github
Copia o conteúdo do arquivo id_rsa.pub nas preferencias do repositório do Github.

(OK) Clonar o projeto em: /home/"user"/projeto
git clone git@github.com:leandrosevenino/algodaodoce.git

(OK) - Criar um link simbólico da pasta /home/"user"/projeto/frontend/under_construction em:
/var/www
Em /var/www:
sudo mv html html_origin
sudo ln -s ~/algodaodoce/frontend/under_construction html

(OK) - Criar um link simbólico da pasta /home/"user"/projeto/frontend/website/dist em:

```
/var/www/demo (Quando em desenvolvimento)
e
/var/www (Quando a versão final estiver OK)
```

(OK) - Criar um link simbólico da pasta /home/"user"/projeto/layout em:

```
/var/www/layout
```

Em /var/www:

```
sudo ln -s ~/algodaodoce/frontend/layout layout
```

Editar o arquivo:

```
/etc/nginx/sites-available/default:
```

```
adicionar:
```

```
location /layout {
    alias /var/www/layout;
}
```

```
sudo service nginx restart
```

f) Subir o SpringBoot com o jetty ou tomcat embarcado (Talvez não precise instalar o tomcat por fora)

Ver a documentação do SpringBoot como rodar ele numa VPS.

<https://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-running-your-application.html>

<https://docs.spring.io/spring-boot/docs/current/reference/html/deployment-install.html>

<https://spring.io/blog/2014/03/07/deploying-spring-boot-applications>

[Para rodar o projeto no servidor]

Ob: Precisa instalar SDKMan antes e depois instalar o Maven com o SDKMain.

No diretório onde está o pom.xml do projeto:

```
$ mvn spring-boot:run
```

testar abrindo o navegador e acessando: <http://dominio.com.br:8080> ou <http://dominio.com.br:8080/api>

Ob: Uma forma bem simples de fazer deploy em VPS:

depois do jar do projeto gerado:

- Criar um link simbólico para o jar do projeto em /etc/inid.d:

```
sudo ln -s $PWD/webapp.jar /etc/init.d/webapp
```

- Botar o serviço a rodar quando o SO iniciar:

```
sudo update-rc.d webapp defaults
```

- sudo service webapp start

- sudo service webapp status

Ob: Para atualizar o projeto:

- sudo service webapp stop

- git pull && mvn package

- sudo systemctl daemon-reload

- sudo service webapp start

- sudo service webapp status

[Perfumarias]

- Ver como pegar no applications.properties o nome do banco, user e senha de variáveis de ambiente

setados no /etc/profile

- Ver a opção de colocar um servidor de cache ecache.

VER:

tomcat

Ver se precisa instalar

<https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-8-on-ubuntu-16-04>

<https://www.digitalocean.com/community/tutorials/how-to-optimize-your-tomcat-installation-on-ubuntu-14-04>

wildfly

Ver se precisa instalar

g) Perfumarias

1. Apontar para um servidor de LOG externo. logz.io e loggly.com

2. Apontar para um servidor de media (Cloudinary.com).

3. Fazer o build e deploy automatico pelo travis-ci.org.

(OK) 4. Ver como configurar as contas de e-mail para usar o dominio no GMAIL.

5. Ver se precisa instalar o FTP ou se o seridor de media já resolve.

6. Melhorias de performace no NGINX:

(OK) Instalar a compactação tar.gz (mod_gzip)

(OK) (Basta descomentar as linhas no arquivo /etc/nginx/nginx.conf)

Instalar e configurar o Varnish

[https://www.google.com.br/search?q=varnish+cache+server&oq=varnish&aqs=chrome..](https://www.google.com.br/search?q=varnish+cache+server&oq=varnish&aqs=chrome..69i57j0l5.3208j0j7&sourceid=chrome&ie=UTF-8#q=digitalocean+varnish)

[69i57j0l5.3208j0j7&sourceid=chrome&ie=UTF-8#q=digitalocean+varnish](https://www.google.com.br/search?q=varnish+cache+server&oq=varnish&aqs=chrome..69i57j0l5.3208j0j7&sourceid=chrome&ie=UTF-8#q=digitalocean+varnish)

Ver como configurar o NGINX para atender múltiplos domínios.

7. Instalar MongoDB ?

8. Instalar Redis ?

9. Ter um CSS para impressão.

10. Ver se funciona com Docker também.

[https://www.google.com.br/search?q=VPS+and+Docker&oq=VPS+and+Docker&aqs=chrome..](https://www.google.com.br/search?q=VPS+and+Docker&oq=VPS+and+Docker&aqs=chrome..69i57j0l5.4972j0j7&sourceid=chrome&ie=UTF-8)

[69i57j0l5.4972j0j7&sourceid=chrome&ie=UTF-8](https://www.google.com.br/search?q=VPS+and+Docker&oq=VPS+and+Docker&aqs=chrome..69i57j0l5.4972j0j7&sourceid=chrome&ie=UTF-8)

postgres://tukwgvzy:JMJEMEwXK5h8LVZk08DUnAq5UvdC-aTt@pellefant.db.elephantsql.com:5432/tukwgvzy

webpack para minificar e comprimir os conteudos

testar em smartphones com o throttle

--

Atenciosamente,

Leandro Severino - Analista Programador.

```
class Leandro(Human, Developer, DevOps):
```

```
    def __init__(self):
```

```
        LinkedIn = https://br.linkedin.com/in/leandroseverino
```

```
        About_me = https://about.me/leandroseverino
```

```
        GTalk = lseverino@gmail.com
```

```
        Skype = leandro-professionalit
```

```
        Github = https://github.com/leandroseverino
```

```
        Slideshare = https://pt.slideshare.net/leandroprofessionalit
```

```
        Blog = http://profissionaldetecnologia.blogspot.com.br
```

```
        Dicas de Tecnologia = https://www.facebook.com/groups/dicas.de.tecnologia.br/
```

```
        Fone_whatsapp = +55 (51) 9.9368.5595 / +55 (51) 9.9390.3511
```



Sender notified by

[Mailtrack](#)