**ESTRUTURA VPS**

Imagem da VPS: **Ubuntu 20.04 LTS**

**INSTALAR AAPANEL**

- Instalar o aapanel: **wget -O install.sh** [**http://www.aapanel.com/script/install-ubuntu\_6.0\_en.sh**](http://www.aapanel.com/script/install-ubuntu_6.0_en.sh) **&& sudo bash install.sh**

- Entrar no **aapanel** -> **App Store** e **instalar**:

**PRIMEIRO O PACOTE LNMP Method = Fast sugerido pelo aapanel! Alguns itens já são instalados pelo LNMP, desta forma, esses itens devem ser ignorados da lista abaixo. A lista abaixo é para listar as versões e os outros que devem ser instalados.**

Node.js version Manager 1.6

Nginx 1.19.7

MySql 5.7.34

PHP-7.4

phpMyAdmin 5.0

PM2 Manager 5.1

SYS Firewall 2.5

Linux Tools 1.7

- Conferir o time zone no **aapanel -> Settings -> Server time**

**INSTALAR DEPENDÊNCIAS DO PUPPETEER**

**sudo apt-get install -y libgbm-dev wget unzip fontconfig locales gconf-service libasound2 libatk1.0-0 libc6 libcairo2 libcups2 libdbus-1-3 libexpat1 libfontconfig1 libgcc1 libgconf-2-4 libgdk-pixbuf2.0-0 libglib2.0-0 libgtk-3-0 libnspr4 libpango-1.0-0 libpangocairo-1.0-0 libstdc++6 libx11-6 libx11-xcb1 libxcb1 libxcomposite1 libxcursor1 libxdamage1 libxext6 libxfixes3 libxi6 libxrandr2 libxrender1 libxss1 libxtst6 ca-certificates fonts-liberation libappindicator1 libnss3 lsb-release xdg-utils**

**INSTALAR O GOOGLE CHROME NO UBUNTU**

Pelo terminal do aapanel:

**wget** [**https://dl.google.com/linux/direct/google-chrome-stable\_current\_amd64.deb**](https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb)

**sudo apt-get install ./google-chrome-stable\_current\_amd64.deb**

**EXECUTAR TODOS OS COMANDOS ACIMA ANTES DE PROSSEGUIR!!! *#início da contrib @grupo3ds*
BANCO DE DADOS**

- Criar o banco de dados através do **aapanel** no menu **Databases, essas informações irão no .env do backend.**

****

**INSTALAR O WHATICKET AAAPANEL**

- Pelo **aapanel -> Files**, ir para o **wwwroot** e clicar no botão **Terminal**.



Com o terminal aberto, digitar: **git clone** [**https://github.com/canove/whaticket/**](https://github.com/canove/whaticket/) **pastaescolhida [whaticket] – pode colocar outro nome, basta considerar o nome dessa pasta em “pastaescolhida”**

****

****

- Dentro do diretório da **pastaescolhida [whaticket]**, subdiretório **backend**, criar o arquivo .**env** com o conteúdo:



***[PODE CRIAR EM* NEW > BLANK FILE *com nome* .env *- ou pelo TERMINAL PELO COMANDO* nano .env *]***



 **COLE O CONTEÚDO:**

NODE\_ENV=DEVELOPMENT

BACKEND\_URL=https://**api**.seudominio.com.br
FRONTEND\_URL=https://**app**.seudominio.com.br
PROXY\_PORT=443
PORT=8080

jwt\_secret=COLE A RESPOSTA DE openssl rand -base64 32 **RODAR esse commando no terminal**



jwt\_refresh\_secret=COLE A RESPOSTA DE openssl rand -base64 32 **RODAR NOVAMENTE esse comando no terminal**

DB\_HOST=localhost

DB\_DIALECT=mysql

DB\_USER=**user**

DB\_PASS=**pwd\_do\_banco**

DB\_NAME=**banco\_name**

JWT\_SECRET=3123123213123

JWT\_REFRESH\_SECRET=75756756756

CHROME\_BIN=/usr/bin/google-chrome-stable **(verificar se instalou com esse nome mesmo)**


**CTRL+S pra salvar**

Após criar o **.env** do **backend**, rodar os comandos dentro do **backend**:


**npm install**

**npm run build**

**npx sequelize db:migrate**

**npx sequelize db:seed:all**

- Dentro do diretório da **escolhido** subdiretório **frontend**, criar o arquivo **.env** com o conteúdo:




CTRL+X | Y | Enter (para salvar caso crie pelo terminal)

CTRL+S (para salvar caso crie pelo painel)

REACT\_APP\_BACKEND\_URL=https://**api**.seudominio.com.br

Ainda dentro do **frontend**, setar a porta para o servidor **frontend** no arquivo **server.js**

****

Depois de definir a porta, rodar o comando no terminal:



npm install

npm run build



**PM2 MANAGER**

DEPOIS DE INSTALAR O PM2 MANAGER pela APP STORE, ABRA-O



CRIE AS TAREFAS (CLICANDO EM **ADD PROJECT**):



PREENCHER MANUALMENTE

OU SELECIONAR NA PASTA

BACKEND:

-Backend Startup file: /www/wwwroot/**whaticket**/backend/dist/server.js

****-Backend Run dir: /www/wwwroot/**whaticket**/backend (**REMOVER** /dist)

-Name: BACK-WHATICKET (qualquer nome, **NÃO USE ESPAÇO**)

-Backend User: **www**

FRONTEND: (MESMO PROCESSO DO BACKEND)

- Frontend Startup file: /www/wwwroot/**whaticket**/frontend/server.js

- Frontend Run dir: /www/wwwroot/**whaticket**/frontend (NADA PARA REMOVER)

- Name: FRONT-WHATICKET (qualquer nome, **NÃO USE ESPAÇO**)

- Frontend User: **www**

 .

**APÓS CRIAR AS DUAS TAREFAS, EXECUTAR MAPPING PARA AS DUAS TAREFAS, ISSO IRÁ CRIAR OS SITES. POR EXEMPLO , GERAR MAPPING PARA apicontratante.com.br e contratante.com.br.**

****

** **

****

Pelo **aapanel -> Files dar permissão 777** na pasta root da contratante e seus subdiretórios. **/www/wwwroot/pastaescolhida**

****

**LIBERAR PORTAS**

- Liberar as portas configuradas no .env e server.js através do aapanel ->Security.





**SSL**

- Instalar pelo **aapanel -> Websites**.



Se for o primeiro certificado para o domínio em questão, fazer a verificação pelo **DNS**.



Caso seja site para o mesmo domínio que já exista o certificado gerado, ir na **TAB Certificate holder** e aplicar o certificado já gerado. Se fizer verificação DNS várias vezes para o mesmo domínio, isso irá gerar vários certificados para o mesmo domínio.

Na configuração do SSL do site, **NÃO** marquei a opção **Force HTTPS**.