

Credenciais:

pocpermuta.kukac.ai

root

uK97568uk

209.172.47.58

pocpermuta.kukac.ai

kadmin

Pocp97568ak

9732

209.172.47.58

Usar comando ssh no prompt ou WinSCP

1* Comando ssh: [kadmin@209.172.47.58](ssh:kadmin@209.172.47.58) -p 9732

2* Informar a senha:

Após informar a senha você estará com acesso kadmin nesta pasta

```
[kadmin@pocpermuta ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
```

3* Volte duas pastas usando:

Cd ..

Cd ..

```
[kadmin@pocpermuta /]$ ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
```

4* Vá para a pasta var/www com o comando:

cd var/www

```
[kadmin@pocpermuta /]$ cd var/www
[kadmin@pocpermuta www]$ ls
cgi-bin clube-permuta html orchestrator teste
```

5* Vá para a pasta do clube-permuta, e pra pasta do respectivo modulo que está trabalhando

Cd clube-permuta

Cd "nome do modulo"

Se preferir pode listar o que tem nas pastas sempre que quiser com o comando
"ls" LS

6* Faça um clone do seu projeto, com o git clone e mantenha atualizado sempre que tiver uma versão instável indo ate a pasta e dando pull no projeto

INSTALAÇÃO DO NODE-MODULES

Com o usuário kadmin você conseguira remover arquivos, mas não criar pastas.

7* Mudando para o usuário root

Digite:

Su

8* Informe a senha do usuário root

```
[kadmin@pocpermuta orchestrator]$ su
Password:
[root@pocpermuta orchestrator]# npm i
```

9* Instalando as dependencias:

npm i

Espere instalar as dependências

CONFIGURANDO AS VARIÁVEIS DE AMBIENTE

10* Comando:

vim .env

abra um editor de texto do Linux, se existir o arquivo, ele abrirá com as variáveis que estão preenchidas se não, abrirá vazio. Copie as variáveis e cole aqui, ou digite todas elas.

```
[root@pocpermuta orchestrator]# vim .env
```

Para inserir aperte I, e vá para o modo INSERT

```
-- INSERT --
```

Para voltar aperte ESC

Para sair salvando digite

```
:wq
```

```
:wq
```

precisamos rodar com pm2

11* precisamos executar o script de build, pois o pm2 roda apenas JS.

```
"build": "babel src --extensions \".js,.ts\" --out-dir dist --copy-files --no-copy-ignored",
```

12* Precisamos ter o babel instalado no projeto

```
"devDependencies": {
  "@babel/cli": "^7.10.1",
  "@babel/core": "^7.10.2",
  "@babel/node": "^7.10.1",
  "@babel/plugin-proposal-class-properties": "^7.14.5",
  "@babel/plugin-proposal-decorators": "^7.15.4",
  "@babel/preset-env": "^7.10.2",
  "@babel/preset-typescript": "^7.10.1",
```

Com o babel e o script corretos e instalados

13* Verifique no tsconfig se o campo esta como a imagem abaixo:

```

"allowJs": true,
"outDir": "./dist",
"rootDir": "./src",
"removeComments": true,
"typeRoots": [
  "./node_modules/@types",
  "./src/@types"
],

```

14* Faça o comando:

npm run build

apos a execução teremos a pasta dist com o projeto em JS

15* agora rodaremos com o comando

pm2 start ./dist/bin/www.js --name nome do projeto

```

[root@pocpermuta orchestrator]# npm run build
> l3hd-sebrae-orchestrator@1.0.0 build
> babel src --extensions ".js,.ts" --out-dir dist --copy-files --no-copy-ignored

Successfully compiled 49 files with Babel (1294ms).
[root@pocpermuta orchestrator]# ls
babel.config.js  jest.config.js  node_modules  package-lock.json  src
dist            manifest.yml    package.json  README.md          tsconfig.json
[root@pocpermuta orchestrator]# cd dist
[root@pocpermuta dist]# ls
app.js  bin  core  documentation  infrastructure
[root@pocpermuta dist]# cd ..
[root@pocpermuta orchestrator]# pm2 start ./dist/bin/www.js --name login
[PM2] Starting /var/www/orchestrator/dist/bin/www.js in fork_mode (1 instance)
[PM2] Done.

```

id	name	namespace	version	mode	pid	uptime	B	status	cpu	mem	user
0	login	default	1.0.0	fork	1085224	0s	0	online	0%	19.0mb	root

```

[root@pocpermuta orchestrator]# pm_

```

Abrir Portas

16* *Firewall-cmd --permanent --zone=public --add-port=3100/tcp*

```

[root@pocpermuta login]# firewall-cmd --permanent --zone=public --add-port=3100/tcp

```

Após esse comando insira o comando de reload

17* *Firewall-cmd --reload*

Obs* Já terá portas abertas, esses comandos são mais por precaução, e conhecimento.

Criando o https

No arquivo [www.js](#)

18* Importe o http e https

```
import http from 'http'|
import https from 'https'
```

19* Crie essa função para criar o https caso já tenha o certificado

```
/**
 * Create HTTP(s) server.
 */

let server: any
if (serverConfig.api.protocol.toLowerCase() === 'https') {
  server = https.createServer({
    key: fs.readFileSync(serverConfig.certificates.sslKey || '', 'utf-8'),
    cert: fs.readFileSync(serverConfig.certificates.sslCert || '', 'utf-8')
  }, app)
} else {
  server = http.createServer(app)
}
```

20* Configurações das variáveis

```
const serverConfig = {
  api: {
    port: process.env.VCAP_APP_PORT || 8080,
    environment: process.env.NODE_ENV,
    protocol: process.env.PROTOCOL || 'http',
    baseUrl: process.env.VCAP_APP_HOST || '0.0.0.0',
    version: process.env.API_VERSION || '1.0.0',
    origin: process.env.API_GATEWAY_HOST || '*',
    vcap: JSON.parse(process.env.VCAP_SERVICES as string || '{}')
  },
  certificates: {
    sslKey: process.env.SSL_KEY,
    sslCert: process.env.SSL_CERT,
    sslChain: process.env.SSL_CHAIN
  },
}
```

Comandos para listar, parar e restartar aplicação no pm2

```
[root@pocpermata login]# pm2 ls
```

id	name	namespace	version	mode	pid	uptime	⌘	status	cpu	mem	user	watching
0	login	default	1.0.0	fork	1095662	13m	0	online	0%	82.1mb	root	disabled

```
[root@pocpermata login]# pm2 stop 0
[PM2] Applying action stopProcessId on app [0](ids: [ '0' ])
[PM2] [login](0) ⌘
```

id	name	namespace	version	mode	pid	uptime	⌘	status	cpu	mem	user	watching
0	login	default	1.0.0	fork	0	0	0	stopped	0%	0b	root	disabled

```
[root@pocpermata login]# pm2 restart 0,
Use --update-env to update environment variables
[PM2][ERROR] Process or Namespace 0, not found
[root@pocpermata login]# pm2 restart 0
Use --update-env to update environment variables
[PM2] Applying action restartProcessId on app [0](ids: [ '0' ])
[PM2] [login](0) ⌘
```

id	name	namespace	version	mode	pid	uptime	⌘	status	cpu	mem	user	watching
0	login	default	1.0.0	fork	1095853	0s	0	online	0%	26.1mb	root	disabled

```
[root@pocpermata login]#
```

pm2 ls – listar aplicações

pm2 stop (id da aplicação) exemplo pm2 stop 0

pm2 restart (id da aplicação) exemplo pm2 restart 0

usando nvm

listar versões do node

nvm ls

```
[root@pocpermata login]# nvm ls
-> v16.13.1
default -> 16.13.1 (-> v16.13.1)
iojs -> N/A (default)
unstable -> N/A (default)
node -> stable (-> v16.13.1) (default)
stable -> 16.13 (-> v16.13.1) (default)
lts/* -> lts/gallium (-> N/A)
lts/argon -> v4.9.1 (-> N/A)
lts/boron -> v6.17.1 (-> N/A)
lts/carbon -> v8.17.0 (-> N/A)
lts/dubnium -> v10.24.1 (-> N/A)
lts/erbium -> v12.22.12 (-> N/A)
lts/fermium -> v14.20.0 (-> N/A)
lts/gallium -> v16.16.0 (-> N/A)
```

Instalar nova versão do node com nvm

nvm i 14.17.6


```

[root@pocpermuta login]# nvm i 14.17.6
Downloading and installing node v14.17.6...
Downloading https://nodejs.org/dist/v14.17.6/node-v14.17.6-linux-x64.tar.xz...
#####
Computing checksum with sha256sum
Checksums matched!
Now using node v14.17.6 (npm v6.14.15)
[root@pocpermuta login]# nvm ls
->      v14.17.6
       v16.13.1
default -> 16.13.1 (-> v16.13.1)
iojs -> N/A (default)
unstable -> N/A (default)
node -> stable (-> v16.13.1) (default)
stable -> 16.13 (-> v16.13.1) (default)
lts/* -> lts/gallium (-> N/A)
lts/argon -> v4.9.1 (-> N/A)
lts/boron -> v6.17.1 (-> N/A)
lts/carbon -> v8.17.0 (-> N/A)
lts/dubnium -> v10.24.1 (-> N/A)
lts/erbium -> v12.22.12 (-> N/A)
lts/fermium -> v14.20.0 (-> N/A)
lts/gallium -> v16.16.0 (-> N/A)
[root@pocpermuta login]#

```

Terá as 2 versões instaladas

Comando para usar uma das versões instaladas

Nvm use 16.13.1

```

[root@pocpermuta login]# nvm use 16.13.1
Now using node v16.13.1 (npm v8.1.2)

```