

 01

Preparando do SDK

Transcrição

Agora que temos o dispositivo virtual pronto e com os arquivos de configurações baixados, podemos seguir para o próximo passo, mas antes, uma observação.

A versão do **Raspbian** utilizada neste curso é a de Setembro de 2017, você pode usar qualquer outra versão desde que se atente a instalar os pacotes que estiverem ausentes em versões anteriores ou posteriores a essa. Você pode baixar a versão mais recente no [site da Raspberry Pi \(https://www.raspberrypi.org/downloads/raspbian/\)](https://www.raspberrypi.org/downloads/raspbian/). Já consideramos que esteja com o mesmo baixado, instalado e em execução para os próximos passos.

Deixamos o arquivo `connect_device_package.zip` na pasta de `Downloads` e precisamos transferi-lo para dentro do Raspberry PI via SSH. Para isso usaremos o terminal e o comando `scp` da seguinte forma:

```
scp connect_device_package.zip pi@192.168.1.5:~
```

Desta forma estamos copiando o arquivo descrito via SSH para o Raspberry PI que está no endereço `192.168.1.5` com o usuário `pi` na pasta `~`. Será pedida a senha de acesso do usuário e logo após isso a cópia é feita.

Uma vez que temos os arquivos copiados para dentro do Raspberry, vamos nos conectar ao mesmo via SSH e realizar o restante das operações. Para conectar ao dispositivo via SSH usamos o comando `ssh` informando o usuário e o endereço.

```
ssh pi@192.168.1.5
```

Informamos a senha novamente e pronto. Estamos conectados!

Por questões de organização, criaremos um novo diretório chamado `aws` com o comando `mkdir` e moveremos o arquivo `.zip` para dentro dele.

```
mkdir aws  
mv connect_device_package.zip aws/
```

Depois do arquivo movido, entraremos na pasta `aws` e o descompactaremos com o comando `unzip`.

```
cd aws  
unzip connect_device_package.zip
```

Podemos usar o comando `ls` para verificar quais arquivos foram descompactados e estes serão os seguintes:

```
rpi-sensor01.cert.pem  
rpi-sensor01.private.key  
rpi-sensor01.public.key  
start.sh
```

Podemos também usar o comando `more` informando o arquivo `start.sh` para termos uma ideia do que está dentro do arquivo e quais passos são executados dentro deste *script*. Essencialmente são 3 passos:

1. Verifica se há um certificado e baixa-o caso não.
2. Instala o Python SDK baixando-o do GitHub caso não esteja instalado. Por isso precisamos do Git Instalado.
3. Iniciar um script de exemplo de publicação e subscrição.

Observação sobre o Python 3 Caso esteja usando o Python 3 como é usado neste curso, talvez precise usar a instrução `sudo` no segundo passo do `start.sh` na seguinte linha `python setup.py install`. Isso por que será feita a instalação do pacote e é preciso de autorização pra isso. A linha passa a ser `sudo python setup.py install`. Você pode fazer isso usando o `vi`. Para sair do `vi` salvando o arquivo lembre-se: Usar o `ESC` para poder digitar comandos e então o comando `wq!`.

Antes de executarmos o script `start.sh` para instalar o SDK, precisamos dar permissão de execução para ele usando o comando `chmod` com as flags `u+x`. `u` para indicar que é o usuário que detém direito sobre o script e `x` para permissão de execução. Lembrando claro de informar para qual arquivo a permissão se aplica.

```
chmod u+x start.sh
```

Python 3 como padrão

Fizemos um comentário anteriormente de que estamos usando o Python 3, ao invés do 2. Você pode usar o 2? Claro! Mas caso queira utilizar o 3 e ainda definí-lo como padrão ao usar o comando `python`, poderá fazer isso da seguinte forma.

Criar um arquivo no home do seu usuário (*o comando `cd` sem parâmetros abre o home do usuário*) chamado `.bash_aliases` e dentro dele ter o seguinte conteúdo:

```
alias python='python3'
```

Dessa forma, ao entrar no `bash` e usar o comando `python`. Será carregado o Python 3 ao invés do 2.

