

stefanoteodorani.it

Home

My Dog

Tags

Categories

Search

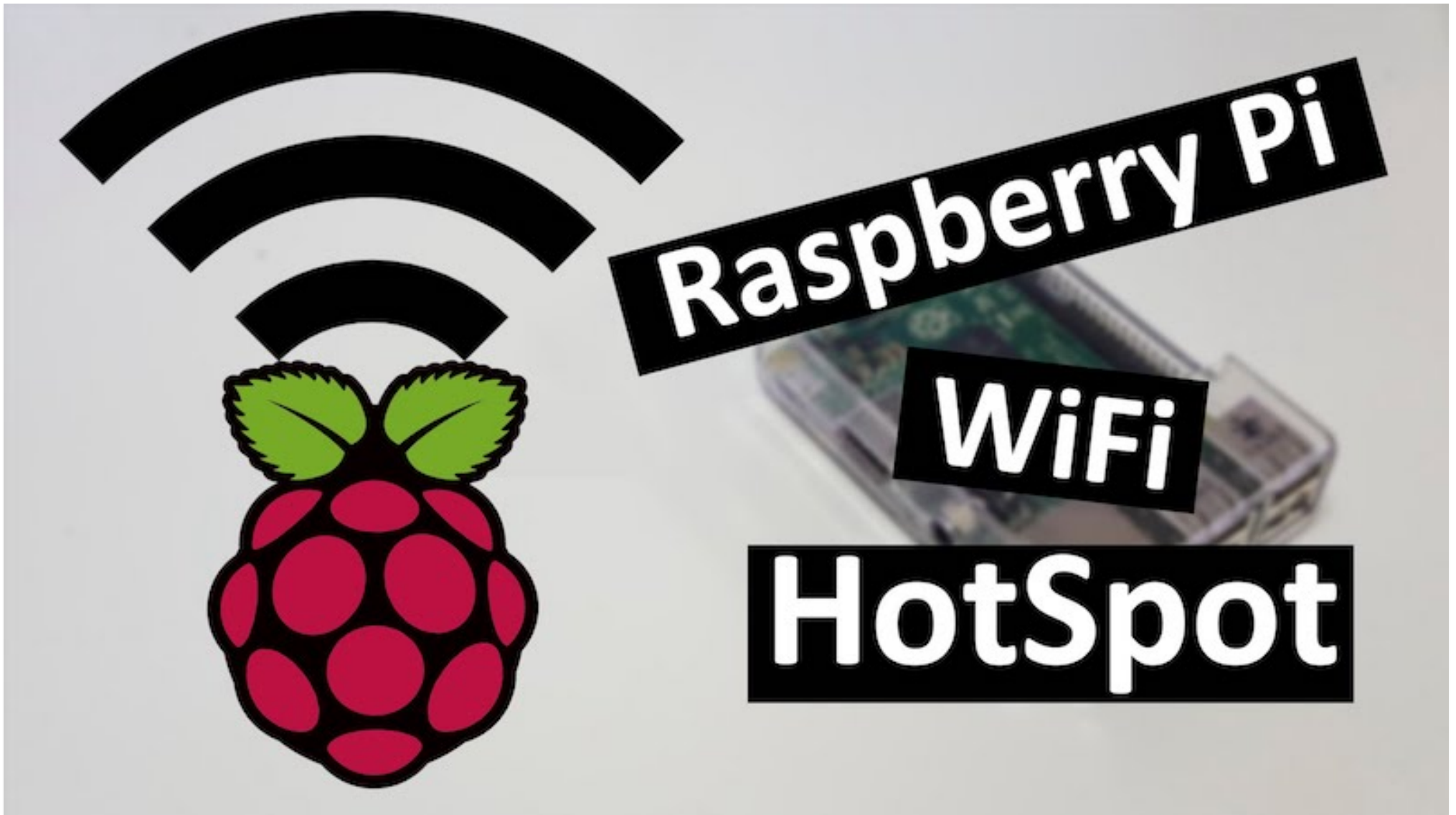


Home

My Dog

Tags

Categories



Hotspot wifi con Raspberry PI 3

📅 20-05-2017 · 878 WORDS · 5 MINUTE READ 📁 FUN 🔖 RASPBERRY · LINUX

La Raspberry PI 3 è stata finalmente dotata di una scheda wifi incorporata. Il chipset della scheda (Broadcom BCM2837) consente di configurare la scheda anche come access point. Non male ! Vediamo come:

Installazione pacchetti

Il primo passo è quello di installare il software necessario. Lo facciamo con il comando:

```
sudo apt-get install dnsmasq hostapd
```

questo installerà:

- hostapd: Consente di utilizzare il WiFi della Raspberry come punto di accesso
- dnsmasq: Questo è un server DHCP e DNS. Semplice da configurare e sufficiente ai nostri scopi.

Configurazione interfacce di rete

La prima cosa da fare è configurare l'interfaccia wlan0 con un IP statico. Ovviamente, per farlo, deve connettersi via rete alla Raspberry. Usa SSH.

Nelle versioni più recenti di Raspian, la configurazione dell'interfaccia viene gestita da dhcpcd per impostazione predefinita. Dobbiamo dirle di ignorare wlan0 , poichè la configureremo con un indirizzo IP statico.

Aprire quindi il file di configurazione dhcpcd con:

```
sudo vi /etc/dhcpcd.conf
```

aggiungere la seguente riga alla base del file:

```
denyinterfaces wlan0
```

Ora dobbiamo configurare l' IP statico. Apri il file di configurazione dell'interfaccia con `sudo vi /etc/network/interfaces` e modifica la sezione wlan0 modo che sia simile a questa:

```
allow-hotplug wlan0
iface wlan0 inet static
    address 172.24.1.1
    netmask 255.255.255.0
    network 172.24.1.0
    broadcast 172.24.1.255
#    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Riavviare dhcpcd con il comando `sudo service dhcpcd restart` e ricaricare la configurazione per wlan0 con `sudo ifdown wlan0` seguito da `sudo ifup wlan0` .

Configurare hostapd

Ora dobbiamo configurare hostapd. Dobbiamo creare un nuovo file di configurazione con il comando `sudo vi /etc/hostapd/hostapd.conf` . ALL'interno dobbiam avere i seguenti contenuti:

```
# This is the name of the WiFi interface we configured above
interface=wlan0

# Use the nl80211 driver with the brcmfmac driver
driver=nl80211

# This is the name of the network
ssid=VoxPopuli-Hub

# Use the 2.4GHz band
hw_mode=g

# Use channel 6
channel=6

# Enable 802.11n
ieee80211n=1
```

```
# Enable WMM
wmm_enabled=1

# Enable 40MHz channels with 20ns guard interval
ht_capab=[HT40][SHORT-GI-20][DSSS_CCK-40]

# Accept all MAC addresses
macaddr_acl=0

# Use WPA authentication
auth_algs=1

# Require clients to know the network name
ignore_broadcast_ssid=0

# Use WPA2
wpa=2

# Use a pre-shared key
wpa_key_mgmt=WPA-PSK

# The network passphrase
wpa_passphrase=raspberry

# Use AES, instead of TKIP
rsn_pairwise=CCMP
```

Ora facciamo una verifica per capire se il lavoro fatto fino a qui funziona correttamente. Eseguiamo quindi:

```
sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

A questo punto, se controlli fra le reti wifi disponibili sul tuo portatile, dovresti essere in grado di vedere la anche rete VoxPopuli-Hub.

Se tentassimo di connetterci a questa rete però, non riusciremmo ad ottenere un IP. Questo perchè ancora non abbiamo configurato `dnsmasq` . Facciamolo. Fai un bel Control-C.

Apri il file di configurazione di hostapd con `sudo vi /etc/default/hostapd` e cerca la riga `#DAEMON_CONF=""` . Deve essere sostituita con `DAEMON_CONF="/etc/hostapd/hostapd.conf"` .

Configurare dnsmasq

Il file di configurazione dnsmasq fornito contiene una grande quantità di informazioni, ma la maggior parte di esse non ci serve. Facciamo quindi una copia del file (non si sa mai) e creiamone uno nuovo.

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
sudo vi /etc/dnsmasq.conf
```

Dentro al nuovo file copiamo quanto segue:

```
interface=wlan0      # Use interface wlan0
listen-address=172.24.1.1 # Explicitly specify the address to listen on
bind-interfaces      # Bind to the interface to make sure we aren't sending things elsewhere
server=8.8.8.8       # Forward DNS requests to Google DNS
domain-needed        # Don't forward short names
bogus-priv           # Never forward addresses in the non-routed address spaces.
dhcp-range=172.24.1.50,172.24.1.150,12h # Assign IP addresses between 172.24.1.50 and 172.24.1.150 with a 12 hour lease time
```

Impostare l' IPV4 forwarding.

Una delle ultime cose che dobbiamo fare è di abilitare l'inoltro dei pacchetti.

Per fare questo apriamo il file `sysctl.conf` con `sudo vi /etc/sysctl.conf` e togliamo il commento dall'inizio della riga che contiene `net.ipv4.ip_forward=1`.

L'operazione appena fatta diventa effettiva al prossimo riavvio della raspberry, ma è possibile attivare subito la modifica con:

```
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

Condividere la connessione internet su eth0

A questo punto facciamo un'ultima operazione che ci consentirà di condividere la connessione internet della Raspberry Pi (attraverso la scheda di rete) con i dispositivi collegati tramite WiFi. Per farlo bisogna configurare un NAT tra l'interfaccia wlan0 e l'interfaccia eth0. Possiamo farlo utilizzando i seguenti comandi:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

Questa ultima configurazione deve essere applicata tutte le volte che la Raspberry viene riavviata, quindi procediamo in questo modo. Prima di tutto ci salviamo le regole in un file:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

poi, per fare in modo che vengano applicate ad ogni reboot della Raspberry, ne aggiungiamo l'esecuzione nel file `rc.local`. Tale file, situato sotto `/etc`, viene eseguito ogni volta che la Rasp parte. Quindi, riassunto:

```
sudo vi /etc/rc.local
```

al cui interno metteremo:

```
iptables-restore < /etc/iptables.ipv4.nat
```

IMPORTANTE: Se non vuoi che la Rasp si blocchi in fase di boot, ricorda di controllare che alla fine del file `/etc/rc.local` ci sia sempre l'istruzione `exit 0` alla fine.

Rush finale

Ci siamo. Facciamo partire i servizi:

```
sudo service hostapd start
sudo service dnsmasq start
```

Ora dovresti essere in grado di connettersi a Internet attraverso la Pi passando dal collegamento WiFi.

Riavvia la raspberry e vedi se tutto continua a funzionare.

Ah! Dimenticavo. Se qualcosa è andato storto, GOTO 10!.

 Comments  Share

OLDER

Oracle grants remapper

NEWER

Digital signage con Pi-Kiosk

RECENTS

WORK

SCHEDULARE NMON NEL CRONTAB

01-05-2020

WORK

PAGINA DI CORTESIA SU APACHE

24-04-2019

WORK

LOCK WAIT E TIMEOUT IN DB2

20-01-2019

WORK

LIQUIBASE VS FLYWAY

22-11-2018

WORK

SQL TUNING ADVISOR PER UNO SPECIFICO SQL_ID

05-11-2018

CATEGORIES

- fun (12)
- work (20)

TAGS

- android (1)
- apache (1)
- ckan (1)
- database (12)

- db2 (1)
- git (1)
- github (1)
- linux (14)
- mac (2)
- mobile (1)
- opendata (1)
- oracle (10)
- raspberry (8)
- ssh (1)
- ubuntu (1)
- utility (4)
- wordpress (1)