Come creare uno script Bash di base per monitorare l'uptime di un sito web

Monitorare l'uptime di un sito web è essenziale per garantire che il tuo sito sia sempre accessibile ai visitatori. Con uno script Bash, puoi automatizzare il controllo dell'uptime e ricevere notifiche immediate in caso di downtime. Ecco una guida passo passo su come creare uno script Bash per monitorare l'uptime di un sito web.

Possiamo creare il seguente codice:

```
#!/bin/bash

# URL del sito web da monitorare
website_url="https://tuosito.com"

# Email per ricevere notifiche
notify_email="tuoemail@example.com"

# Intervallo di tempo tra i controlli (in secondi)
delay_check=60

# Funzione per controllare l'uptime del sito
check_website() {
    local status_code=$(curl -o /dev/null -s -w "%{http_code}\n" "$website_url")
    local timestamp=$(date "+%Y-%m-%d %H:%M:%S")

    if [ "$status_code" -ne 200 ]; then
```

```
    echo "$timestamp - $website_url down. Status
code: $status_code" | mail -s "Down" "$notify_email"
else
    echo "$timestamp - $website_url up. Status code:
$status_code"
fi
}

# Loop infinito per controllare il sito ad intervalli
regolari
while true; do
    check_website
    sleep $delay_check
done
```

Questo script inizierà a monitorare l'uptime del sito web specificato e invierà una notifica email ogni volta che il sito risulterà non raggiungibile (ovvero quando il codice di stato HTTP non è 200).

Per rendere il file eseguibile, digita il seguente comando nel terminale:

```
chmod a+x check-website.sh
```

Puoi personalizzare diversi aspetti dello script:

1. **URL del sito web:** Modifica la variabile `website_url` con l'indirizzo del sito che desideri monitorare.
2. **Email per le notifiche:** Cambia la variabile `notify_email` con il tuo indirizzo email.
3. **Intervallo di tempo:** Regola la variabile `delay_check` per modificare la frequenza dei controlli (in secondi).

Puoi anche eliminare completamente il loop infinito e gestire l'intervallo tra un controllo e l'altro utilizzando un cronjob. Quindi lo script verrebbe modificato come segue:

```
#!/bin/bash

# URL del sito web da monitorare
website_url="https://tuosito.com"

# Email per ricevere notifiche
notify_email="tuoemail@example.com"

# Funzione per controllare l'uptime del sito
check_website() {
    local status_code=$(curl -o /dev/null -s -w "%
{http_code}\n" "$website_url")
    local timestamp=$(date "+%Y-%m-%d %H:%M:%S")

    if [ "$status_code" -ne 200 ]; then
        echo "$timestamp - $website_url down. Status
code: $status_code" | mail -s "Down" "$notify_email"
    else
        echo "$timestamp - $website_url up. Status code:
$status_code"
    fi
}

check_website
```

Digitando quindi `crontab -e`, potrai inserire la seguente riga:

```
* * * * * /bin/bash /usr/local/bin/check-website.sh >  
/dev/null 2>&1
```

Conclusione

Monitorare l'uptime del tuo sito web è cruciale per garantire la disponibilità del servizio ai tuoi utenti. Con un semplice script Bash, puoi automatizzare questo processo e ricevere notifiche immediate in caso di problemi. Questo ti permette di intervenire rapidamente per risolvere eventuali downtime, migliorando l'affidabilità del tuo sito web.