# Creare uno script Bash per monitorare un'applicazione Node.js

In questo articolo, vedremo come creare uno script Bash per monitorare un'applicazione Node.js e riavviare il servizio di sistema associato solo quando i file dell'applicazione vengono modificati. Utilizzeremo `inotifywait` dal pacchetto `inotify-tools` per osservare le modifiche ai file e `systemctl` per riavviare il servizio.

Prima di iniziare, assicurati di avere installato `inotify-tools` e di gestire la tua applicazione Node.js con un servizio systemd.

```
sudo apt-get install inotify-tools
```

Assicurati che la tua applicazione Node.js sia gestita da `systemd` e che tu abbia un file di servizio, ad esempio `myapp.service`.

Crea quindi un file chiamato `monitor.sh` e rendilo eseguibile:

```
touch monitor.sh
chmod +x monitor.sh
```

Apri `monitor.sh` con il tuo editor di testo preferito e inserisci il seguente contenuto:

```
#!/bin/bash
```

```
# Directory da monitorare
MONITOR_DIR="/path/to/your/nodejs/app"

# Nome del servizio
SERVICE_NAME="myapp.service"

# File di log per lo script
LOG_FILE="/var/log/monitor.log"

# Funzione per registrare i messaggi
log_message() {
    echo "$(date +'%Y-%m-%d %H:%M:%S') - $1" >> "$LOG_FILE"
}

log_message "Inizio monitoraggio file per $MONITOR_DIR"

# Funzione per riavviare il servizio
restart_service() {
    log_message "Modifiche rilevate in $MONITOR_DIR. Riavvio
di $SERVICE_NAME."
    systemctl restart $SERVICE_NAME
    if [ $? -eq 0 ]; then
        log_message "$SERVICE_NAME riavviato con successo."
    else
        log_message "Errore nel riavvio di $SERVICE_NAME."
    fi
}

# Gestione dei segnali SIGINT e SIGTERM per pulire prima di
uscire
trap 'log_message "Arresto dello script di monitoraggio.";
exit 0' SIGINT SIGTERM

# Monitorare la directory per le modifiche
inotifywait -m -r -e
modify,attrib,close_write,move,create,delete "$MONITOR_DIR" -
-format '%w%f' |
while read FILE; do
    log_message "File modificato: $FILE"
    restart_service
done
```

Spiegazione dello script:

- **Variabili**:
  - `MONITOR_DIR`: La directory della tua applicazione Node.js.
  - `SERVICE_NAME`: Il nome del servizio systemd che gestisce la tua applicazione.
  - `LOG_FILE`: Un file per registrare i messaggi dello script.
- **Funzioni**:
  - `log_message`: Registra messaggi con timestamp in `LOG_FILE`.
  - `restart_service`: Registra un messaggio, riavvia il servizio e registra se il riavvio ha avuto successo.
- **Gestione dei Segnali**:
  - Lo script gestisce i segnali `SIGINT` e `SIGTERM` per assicurarsi che registri un messaggio quando viene arrestato.
- **inotifywait**:
  - `inotifywait` monitora la directory in modo ricorsivo (`-r`) per eventi specifici: modifica, cambiamento attributi, chiusura scrittura, spostamento, creazione e cancellazione (`-e modify,attrib,close_write,move,create,delete`).
  - L'opzione `--format '%w%f'` manda in output il percorso completo del file modificato.
  - L'output di `inotifywait` viene letto riga per riga. Per ogni riga, lo script registra la modifica e riavvia il servizio.

Puoi eseguire lo script direttamente in background:

```
./monitor.sh &
```

Puoi creare un file di servizio systemd per gestire lo script di monitoraggio. Crea un file chiamato `monitor.service` e inserisci il seguente contenuto:

```
[Unit]
Description=Monitoraggio Modifiche File per App Node.js

[Service]
ExecStart=/path/to/monitor.sh
Restart=always

[Install]
WantedBy=multi-user.target
```

Abilita e avvia il servizio di monitoraggio:

```
sudo systemctl enable monitor.service
sudo systemctl start monitor.service
```

## Conclusione

Seguendo questi passaggi, puoi creare uno script Bash che monitora i file della tua applicazione Node.js e riavvia automaticamente il servizio associato quando vengono rilevate modifiche. Questo è utile per ambienti di sviluppo e produzione, in cui è necessario applicare automaticamente le modifiche senza intervento manuale.