

GABRIELE ROMANATO

Debugging dei cronjob su server Linux

I cronjob sono fondamentali per automatizzare operazioni ricorrenti su sistemi Linux. Tuttavia, possono essere difficili da debuggare quando non funzionano come previsto. Questo articolo illustra alcune tecniche utili per identificare e risolvere problemi comuni con i cronjob.

1. Verificare il file crontab

Per visualizzare i cronjob dell'utente corrente:

```
crontab -l
```

Per modificare i cronjob:

```
crontab -e
```

2. Controllare la sintassi

Assicurati che la sintassi del cronjob sia corretta. Ogni voce ha 5 campi temporali seguiti dal comando:

```
* * * * * /path/to/script.sh
```

3. Log di sistema

Controlla i log di sistema per vedere se i cronjob vengono eseguiti:

```
grep CRON /var/log/syslog
```

Su sistemi basati su RHEL/CentOS:

```
grep CRON /var/log/cron
```

4. Output di errore

I cronjob non inviano automaticamente l'output a un terminale. Per catturare l'output standard e di errore, usa la redirectione:

```
* * * * * /path/to/script.sh >/tmp/output.log 2>&1
```

5. Variabili di ambiente

I cronjob usano un ambiente molto limitato. Assicurati di impostare le variabili necessarie, come PATH:

```
PATH=/usr/bin:/bin  
* * * * * /path/to/script.sh
```

6. Permessi

Verifica che lo script sia eseguibile:

```
chmod +x /path/to/script.sh
```

7. Test manuale

Esegui manualmente lo script per vedere se funziona al di fuori del contesto di cron:

```
/path/to/script.sh
```

8. Utilizzare un wrapper

Può essere utile eseguire lo script tramite uno script wrapper che logga data e output:

```
#!/bin/bash

echo "Esecuzione: $(date)" >> /tmp/cron_debug.log
/path/to/script.sh >> /tmp/cron_debug.log 2>&1
```

Conclusione

Il debugging dei cronjob richiede attenzione ai dettagli. Controllare i log, reindirizzare l'output e testare manualmente sono tecniche chiave per individuare i problemi. Una volta identificati, i problemi sono spesso semplici da risolvere.