

Bash skripting

Gruppe 20

av

Pål, Ola og Kristoffer

i

IKT106

Nettverksadministrasjon

Veiledet av Erlend Fredriksen

Fakultet for teknologi og realfag

Universitetet i Agder

Grimstad, Mars 2020

Innhold

1	backup skript	1
1.1	Oppsett av Slack	1
1.2	Oppsett av script	1
2	Crontab	2
3	Konklusjon	3

Figurer

1	started	1
2	Rommet	1
3	Slack message	2
4	Cronjob	2

Listinger

1	msg løysing	2
---	-----------------------	---

Introduksjon

I denne oppgaven skal vi lage et skript som automatisk tar backup av en valgt mappe, for så og legge den nye komprimerte filen inn i en ny mappe. vi skal også ta outputen som scriptet gir og sende det til ei fil.log. denne outputen skal og sendes til slack.

1 backup skript

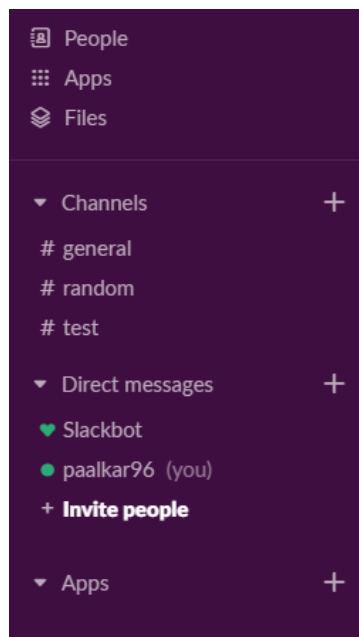
1.1 Oppsett av Slack

oppsetting av slack var ganske rett fram gå innpå slack.com og trykk på “GET STARTED”



Figur 1: started

Så fyller du ut da så må fyllast. Når du har komnt inn i på rommet du har laget trykker du på “Apps”



Figur 2: Rommet

og søker opp “incoming Webhooks” og laster ned det.

1.2 Oppsett av script

Første jeg begynte med var å lage mappene som backup filen og loggen skulle inni. så valgte vi å ta backup av Pictures mappen.

vi bestemte oss for å bruke kun YYYY/MM/DD isteden for å legge til klokkeslett i tillegg, med tanke på at det skal kun tas backup annenhver dag.

vi brukte også kommanden “hostname -s” sånn at vi kan differensiere hvem som har tatt backup tillegg.

hadde en del problem med postinga av outputen til slack. forstod ikke helt hvordan det skulle gjøres, men fikk det til med å bruke det som står i “msg løsning”

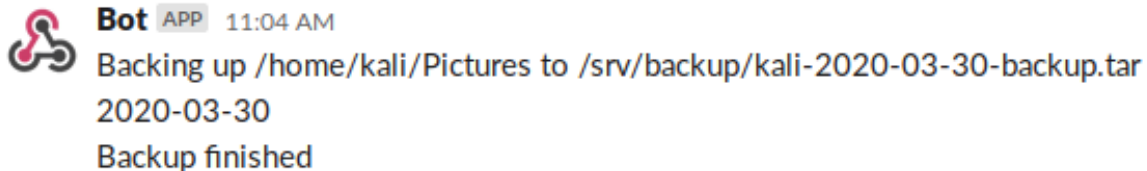
```
#!/bin/bash
```

```
slack_text="${backing_up}\n ${day}\n ${backup_fi}"
```

```
slack_url="https://hooks.slack.com/services/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"
```

```
curl $slack_url -X POST -H "Content-type: application/json" -d "{\"text\": \"$slack_text\"}"
```

Listing 1: msg løysing



Figur 3: Slack message

endret litt på koden og brukte heller “date” for printing av datoen.

2 Crontab

Brukte crontabguru for å finne tidspunkta og dagene vi skulle backup. så gjekk vi inni crontab mappa og satte inn tidspunkt og kommanden den skulle kjøre

```
kali@kali:~/bashscript$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
30 1 * * 1,3,5 kali /home/kali/bashscript/backup-script.sh
```

Figur 4: Cronjob

vi endret cronjoben litt etter dette og inkluderte en `> /var/log/ikt106.log` sånn at outputen på scriptet blir sendt til loggen. Man kan også logge ved å bruke “`set -x » /var/log/ikt106.log`” i begynnelsen av scripte og “`set +x`” i slutten og da logger det outputen og alle linjene i scriptet automatisk inn i `ikt106.log`, men vi valgte heller og gjøre det uten for.

3 Konklusjon

oppretting av slack og cronjob var ganske rett frem og tok ikke lang tid å forstå. scriptet derimot var litt utfordring. Kan sterkt konkludere med at posting av outputs til slack er en bitch om du ikke vet helt hva du driver med.