Lektionstillfälle 15

Logghantering och felsökning
med Mikael Larsson

NACKADEMIN

Återblick

Förra lektionen jobbade vi med:

- tar, gzip, zip
- rsync
- rclone

Dagens lektion

Mål: Att veta hur loggar hanteras och hur man kan undersöka feltillstånd.

- Roterade loggar
- Diskutrymme
- Minne
- CPU
- Öppna filer
- Hängande processer

Termer och begrepp

watch
sysctl
CPU-hog
zombie

Loggar, loggar, loggar

Processer som loggar kontinuerligt kan producera stora mängder.

För att hålla enskilda loggfiler hanterliga brukar man rotera loggarna.

Det innebär att loggfiler döps om och kanske komprimeras med jämna mellanrum.

Exempel:
syslog
syslog.1
syslog.2.gz
...

***Tips**: less kan läsa gzippade textfiler utan uppackning först.*

Centraliserad loggning

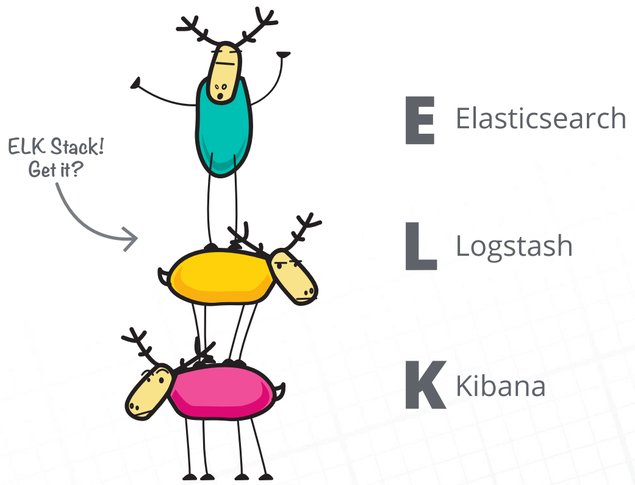
”rsyslog” är det system för centraliserad loggning som kommer med Ubuntu server.

https://www.howtoforge.com/how-to-setup-rsyslog-server-on-ubuntu-1804

Istället för att logga allt direkt på nuvarande dator skickas loggarna till en logg-server.

Det finns också många externa system och tjänster som används till centraliserad loggning.

En populär variant är ELK-stacken, bestående av Elasticsearch, Logstash och Kibana:
https://www.elastic.co/what-is/elk-stack



journalctl

En komponent av **systemd** är **journald**, loggningstjänsten.

Vi har hittills lärt oss generella verktyg för att titta på alla slags loggar, i form av filer på disk.

För att direkt titta i ”journalen” använder man **journalctl**.

journalctl är ett mer komplext verktyg, men är också mer kraftfullt. Det kan ge ett mer precist resultat än att bara greppa i textfiler.

https://blog.selectel.com/managing-logging-systemd

watch

När man övervakar ett system kan det vara bra att få en kontinuerlig uppdatering från ett kommando.

Några kommandon har uppdatering inbyggt, som vi tidigare sett.

Man kan också använda ”watch” som upprepat kör ett kommando för att visa hur det som bevakas ändras över tid.

stress

För att ha något att bevaka på våra små VM:ar kommer vi stressa systemet.

Det finns några verktyg för ändamålet, bl.a. ”stress” och ”stress-ng”.

Det kan också vara ”kul” att själv sätta ihop stresstester.

https://bash-prompt.net/guides/create-system-load

ulimit

För att ett Linux-system inte ska lastas för hårt av enskilda användare, eller alla användare tillsammans, finns det olika begränsningar.

Begränsningar för användaren hanteras med ”ulimit”.
För att visa vilka limits som är aktiva:

```
$ ulimit -a
```

Man kan ställa in limits för systemet i ”/etc/security/limits.conf”.

Diskutrymme

För att bevaka utnyttjat och ledigt utrymme på alla filsystem används ”df”, som vi tidigare sett

Man kan också använda ”du”, ***disk usage***, för att se hur mycket plats en katalog eller ett filträd tar.

Vanliga flaggor till ”du” är ”-sh”, som inkluderar underkataloger (Subdirectories) och visar storleken i ”Human” – enheter.

```
$ du -sh /home/frasse
```

CPU

Som vi tidigare sett använder vi **top** eller **htop** för att övervaka CPU och till viss del minneslast på ett system.

Båda verktygen visar lastsiffror i form av tre tal:

```
load average: 0.00, 0.02, 0.00
```

Det här är det klassiska ”load” – värdet och består av medelvärdet av systemets CPU-**behov** de senaste 1, 5 och 15 minuterna.

Behovet motvaras av antalet trådar som just nu kör eller väntar på att få köra.

Om siffran är högre än antalet CPU-er är systemet överutnyttjat.

http://www.brendangregg.com/blog/2017-08-08/linux-load-averages.html

Minne

Minne övervakas med **free** men även med **top/htop**:

$ free –h

Man kan även använda ”vmstat”, som även ger en bild av cpu-last och i/o.

$ vmstat

OBS: vmstat visar som standard värden sedan senaste boot, vilket ofta inte är så intressant.

Anger man en siffra (antal sekunder) till **vmstat** kommer den visa statistik upprepat för det intervallet.

Det finns en speciell kärn-tjänst som bevakar minnesanvändning och dödar processer för att frigöra minne: ***oom_reaper.***

Ibland i system med hård minneslast kan man se att i syslog ***oom_reaper*** *har klivit in.*

I/O

vmstat är också bra för att se last på diskar i form av läsningar och skrivningar.

iotop funkar som **top**, men visar istället I/O. Mycket användbart för att leta rätt på processer som lastar disk eller nätverk.

Öppna filer

Vi har tidigare använd både **lsof** för att undersöka vilka filer som är öppna och av vem.

För att felsöka kan man lista

- filer per användare:

```
$ lsof -u frasse
```

- filer per process:

```
$ lsof -p 4711
```

Hängande processer

Om man har en process som verkar hänga kan man såklart avbryta den.

Men – det kanske är ett jobb där man förlorar resultatet om den bara avbryts.

Istället vill man ta reda på vad processen håller på med. Då använder man ”strace”.

strace, system trace, visar vilka systemanrop en process gör och vilka tillstånd den går igenom.

```
$ strace -p 4711
```

Det finns också zombie-processer: https://www.howtogeek.com/119815/htg-explains-what-is-a-zombie-process-on-linux

kill

När en process ska avslutas använder man ”kill”.
Om inte signalen som ”kill” ska skicka anges, skickar den signal 15, SIGTERM.

Processer som hänger på riktigt avslutas inte av SIGTERM. Då behöver man ta i med hårdhandskarna, d.v.s. SIGKILL eller 9.

”kill -9” är lite som processernas ”sudo” – det går inte att värja sig. Det är inte processen själv som hanterar SIGKILL, utan kärnan avslutar processen direkt.

*Istället för att själv pajpa och greppa kan man använda **pgrep** för att leta rätt på processer.*

stop + continue

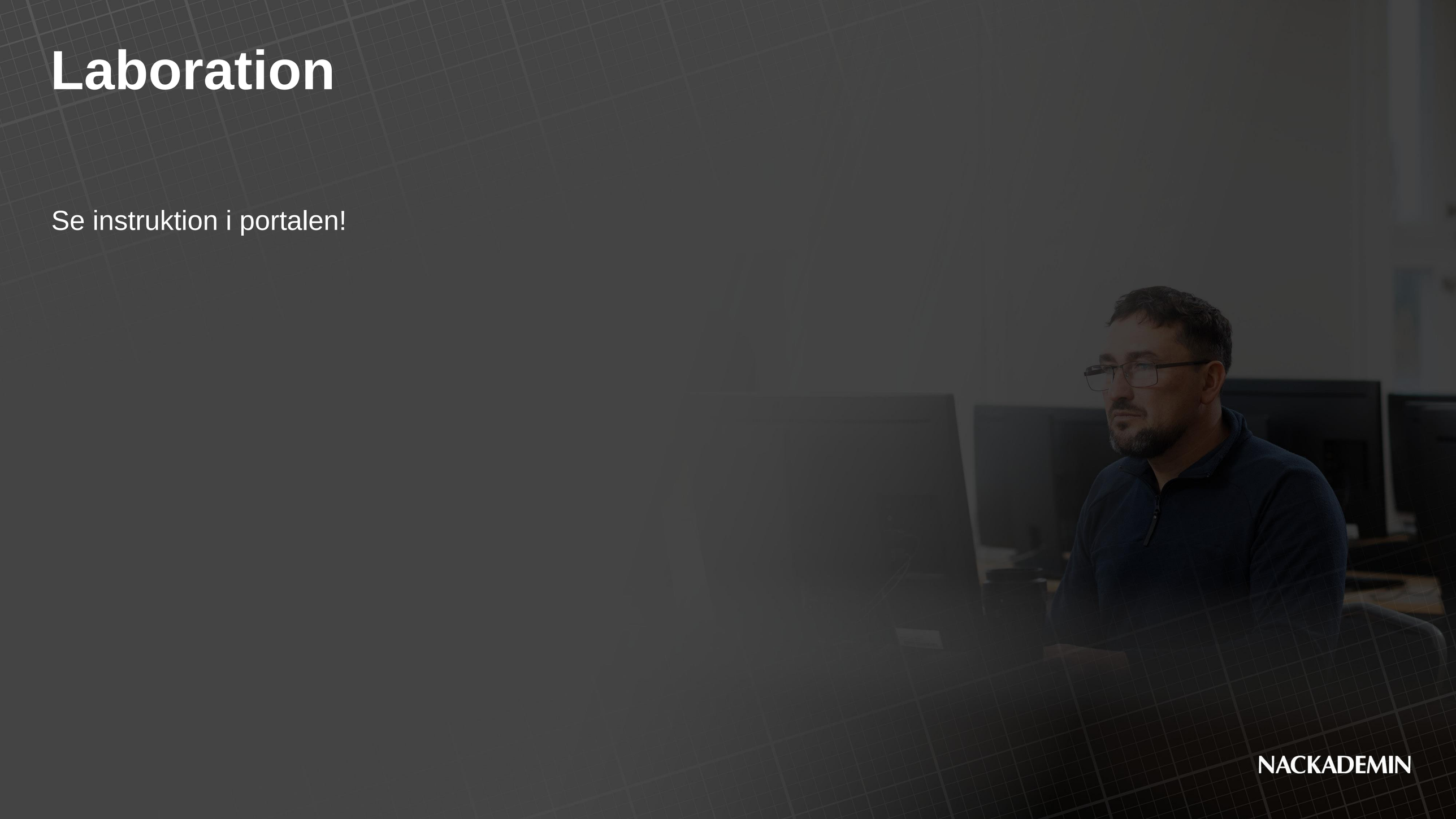
Om man har en process som sprungit iväg, och man vill pausa en stund kan man använda signalerna SIGSTOP och SIGCONT.

Det kan till exempel vara praktiskt om en process håller på att fylla disken och du behöver tid att rensa upp.

https://major.io/2009/06/15/two-great-signals-sigstop-and-sigcont

Laboration

Se instruktion i portalen!



NACKADEMIN

Önskemål om repetition

I övermorgon kommer vi ägna oss åt repetition och fördjupning

Om ni redan nu känner att det är några områden ni vill repetera eller veta mer om, **säg till!**

Det underlättar min planering.
Tack!

Summering

Idag har vi tittat på olika verktyg för att titta på olika aspekter av systemlast och hitta problem.

Vi har också tittat på några sätt att lösa lastproblem.

Nästa gång

Mål: Installera grafiskt gränssnitt i Ubuntu server, och anpassa utseendet i bash.

- X Window System (X11, X)
- .bashrc, .profile, alias
- Planering av repetition
- **Sista inlämningsdag inlämningsuppgifter!**

Stort tack!

NACKADEMIN