how to setup logrotation and dstat monitoring

This describes how logrotation and dstat monitoring has been setup for each linux VM in PTN environment.

logrotation section can be applied to any process that involves writing regularly to a logfile.

Why logrotation?

if a logfile is written to regularly, its size will grow quickly. This presents a problem if a user needs to access information written in the logfile as it can take a long time to grep or split into smaller manageable files.

The answer to this is to use logrotation. Logs can be rotated on daily, weekly, monthly basis etc. The logs in this case are written to the local filesystem, but then can be moved elsewhere for archiving.

Setup logrotation

By default, linux has logrotation setup for the file rsyslog (which logs system events, exceptions etc). So its easier to add a new file to be logrotated to the rsyslog config file.

edit /etc/logrotate.d/rsyslog to add section on dstat log (insert after syslog section):

```
/root/dstat.csv
{

rotate 4
daily
copytruncate
compress

}
```

using dstat as an example but any file can be added. There are a number of actions that can be taken in logrotation. The above instructs logrotation to:

- rotate (ie keep) last 4 files
- rotate on a daily basis
- truncate original file after copying
- compress (gzip) the file to be saved

NB there are other actions that can be used.

setup cronjobs

Now it is important to set the cronjobs that correspond to the logrotation

There are several cronjobs that run, as this is a daily logrotation, need to specify the time to execute.

edit /etc/crontab:

change cron.daily to run at midnight, ie
00 0 * * * root test -x /usr/sbin/anacron || (cd / && run-parts --report /etc/cron.daily)

you can choose any time you wish ,but bear in mind that cron.daily job by default runs at 6am.

Now also, for a process that writes to a logfile it is important to stop the process before running logrotation otherwise the log will not be truncated. In the case of dstat, there are 2 scripts that stop and start the dstat process (discussed later). These also need to be setup as cron jobs.

sudo crontab -e:

add the following entries to stop and then start dstat process.

```
59 23 * * * /root/dstat-stop.sh >/root/cron-debug.log 2>&1
02 00 * * * /root/dstat-start.sh >/root/cron-debug.log 2>&1
```

For this , it is important to give enough time for logrotation to execute fully. If files are very large it may take time to compress then copy, truncate etc

start/stop scripts

The following is specific to dstat but can be applied to any other process. If the process is not stopped before logrotation, the original file will keep on growing and the purpose of using logrotation will be lost.

For PTN environment, I decided best implementation is to create 2 scripts to control the stopping and startup of dstat.

These scripts are placed in /root and are referred to in 2nd cronjob.

contents of these scripts are:

dstat-start.sh

#!/bin/bash

#startup dstat after logrotation

dstat --cpu -C all -lmnp --top-cpu -d --top-mem -t --output dstat.csv &>/dev/null &

there are several parameters supplied to dstat, including output file (log file that will be rotated). NB &> is to suppress the output to the screen as dstat by default writes to the terminal even if output file is specified.

dstat-stop.sh

#!/bin/bash

#kill dstat process before logrotation

kill -9 $(ps -ef | grep 'python2 /usr/bin/dstat' | grep 'root' | grep -v 'color=auto' | grep -v 'grep' | awk '{print $2;exit}')

the stop script has to kill the particular pid of dstat, so its a bit complex. Previously this script was simpler but failed to kill the dstat process and so there were multiple dstat versions running which was not good.

Once successfully running, there will be up to 4 daily backup files for dstat logs plus the current file: