Protecting identities with intelligent authentication

swivelsecure

Free MemCache Script creation and cron job

Swivel Secure Limited
Punto Mobi 4 28850 Alcala de Henares, Madrid, Spain

www.swivelsecure.com

Contents

Version History

Release Date	Description	Version	Author
30/12/2021	Initial Creation	V1.0	Pablo Garcia

Intellectual Property Rights

This document may contain valuable trade secrets and confidential information of Swivel Secure and its partners and customers, and shall not be disclosed to any person, organization, or entity unless such disclosure is subject to the provisions of a written non-disclosure and proprietary rights agreement or intellectual property license agreement approved by Swivel Secure.

Protecting identities with intelligent authentication

swivelsecure

Pre-requisites

This list defines the pre-requisites necessary to perform this Documentation.

#	Description
1	Linux OS commands
2	Perl Script Code
3	Webmin Config

This document explains the code to be implemented into a Perl script which consist of a pair of OS commands to free MemCache at OS Level.

This execution increases available RAM memory and is recommended high load environments with limited/constrained resources.

Useful commands

`free -h`

```
[root@single ~]# free -h
             total       used       free     shared    buffers     cached
Mem:          7.8G       7.6G       204M        16K       234M       6.1G
-/+ buffers/cache:       1.3G       6.5G
Swap:         1.0G        43M       980M
```

`top -S -d 0.1`

```
[root@single ~]# top -S -d 0.1


top - 18:29:20 up 26 days,  7:30,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 150 total,   1 running, 149 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  0.0%sy,  0.0%ni,100.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   8176280k total,  7967524k used,   208756k free,   240036k buffers
Swap:  1048572k total,    44404k used,  1004168k free,  6351656k cached

   PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 83025 root      20   0 15024 1364 1008 R 18.9  0.0   0:00.35 top
     1 root      20   0 19356 1248 1048 S  0.0  0.0  56:21.37 init
     2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
     3 root      RT   0     0    0    0 S  0.0  0.0   0:01.25 migration/0
     4 root      20   0     0    0    0 S  0.0  0.0   0:10.13 ksoftirqd/0
     5 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 stopper/0
     6 root      RT   0     0    0    0 S  0.0  0.0   0:02.99 watchdog/0
     7 root      RT   0     0    0    0 S  0.0  0.0   0:01.65 migration/1
     8 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 stopper/1
     9 root      20   0     0    0    0 S  0.0  0.0   0:05.93 ksoftirqd/1
    10 root      RT   0     0    0    0 S  0.0  0.0   0:02.07 watchdog/1
    11 root      RT   0     0    0    0 S  0.0  0.0   0:01.25 migration/2
    12 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 stopper/2
    13 root      20   0     0    0    0 S  0.0  0.0   0:15.83 ksoftirqd/2
    14 root      RT   0     0    0    0 S  0.0  0.0   0:02.06 watchdog/2
    15 root      RT   0     0    0    0 S  0.0  0.0   0:01.94 migration/3
    16 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 stopper/3
```

0.1 is the info refresh time (default 1 sec.)

These commands print Memory Status at **OS level**.

TOTAL RAM USED RAM FREE RAM TOTAL SWAP USED SWAP CACHED MEM

Top also shows current time; uptime in days and hours, minutes; CPU %load; PID, %CPU, %MEM of consumption of processes.

Free cached memory results in free RAM too and can help to the whole system to overcome high load peaks if it is programmatically set up.

Not all systems could need this but is recommended if resources are fair in contrast to the use of the product.

Script code and creation

The script is a perl execution of two OS commands that free the CachedMem at OS level.

CODE

```
#!/usr/bin/perl -w
use strict;

my $cmd = "sync && sysctl -w vm.drop_caches=3";
my $result = `$cmd`;
print $result;
```

1. Log in to the OS console via CMI

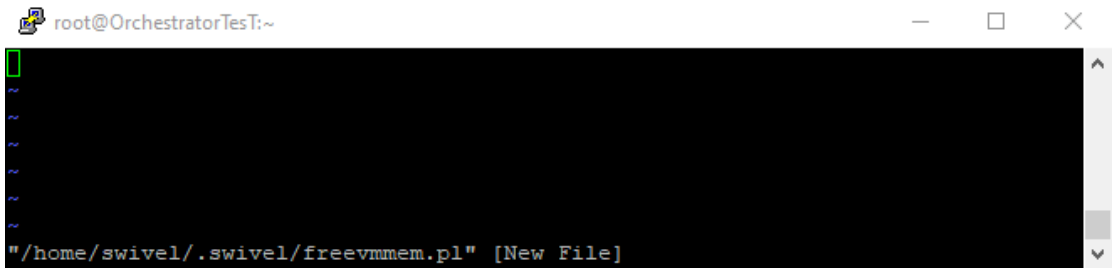
```
To continue to the Linux command prompt please enter the CMI password or press
"Return" to return back to the menu

Password:
INFO: Command Line accessed
Type 'exit' to return to CMI

[root@single ~]#
```

2. Create a file in /home/swivel/.swivel/freevmmem.pl

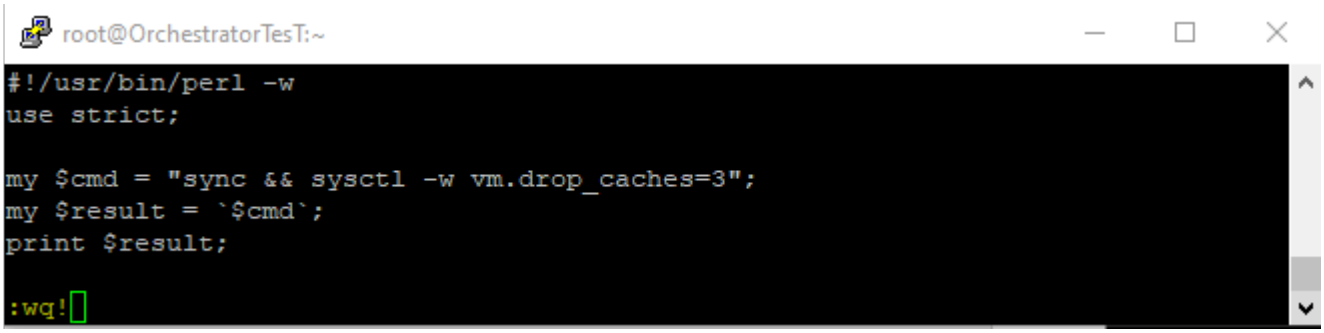
```
[root@single ~]#vi /home/swivel/.swivel/freevmmem.pl
```



3. Press 'A' button to present *INSERT*

4. Copy the *CODE* and paste (right button)

5. Save the file (*ESC* key + *:wq!* and *ENTER*)



6. File ownership and permissions

```
[root@single ~]#ls -la /home/swivel/.swivel/freevmmem.pl
-rw-r--r-- 1 root root 117 Dec 29 18:55 /home/swivel/.swivel/freevmmem.pl

[root@single ~]# chown swivel:swivel /home/swivel/.swivel/freevmmem.pl

[root@single ~]# chmod 755 /home/swivel/.swivel/freevmmem.pl

[root@single ~]# ls -la /home/swivel/.swivel/freevmmem.pl
-rwxr-xr-x 1 swivel swivel 117 Dec 29 18:55 /home/swivel/.swivel/freevmmem.pl
```

7. At this point, script can be executed manually,

```
[root@single ~]# sudo perl /home/swivel/.swivel/freevmmem.pl
vm.drop_caches = 3
[root@single ~]#
```

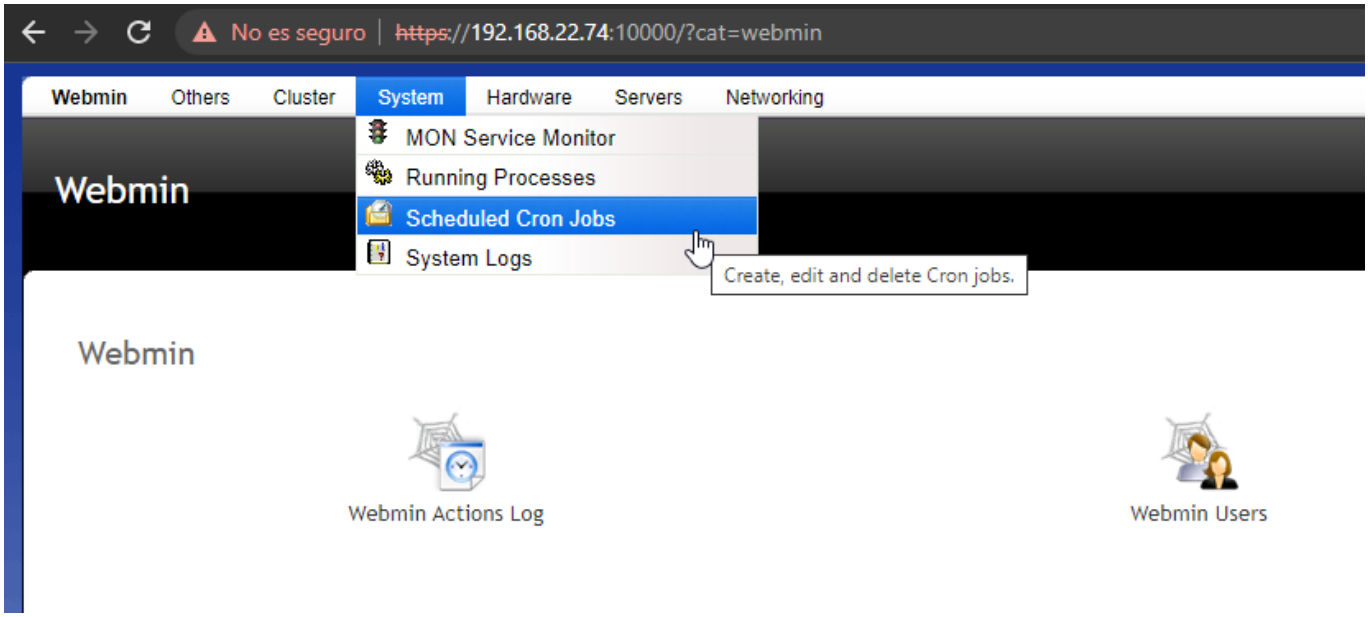
But if the objective is executing this script programmatically, this must be done using a cron job which is configured using Webmin.

Webmin cronjob config

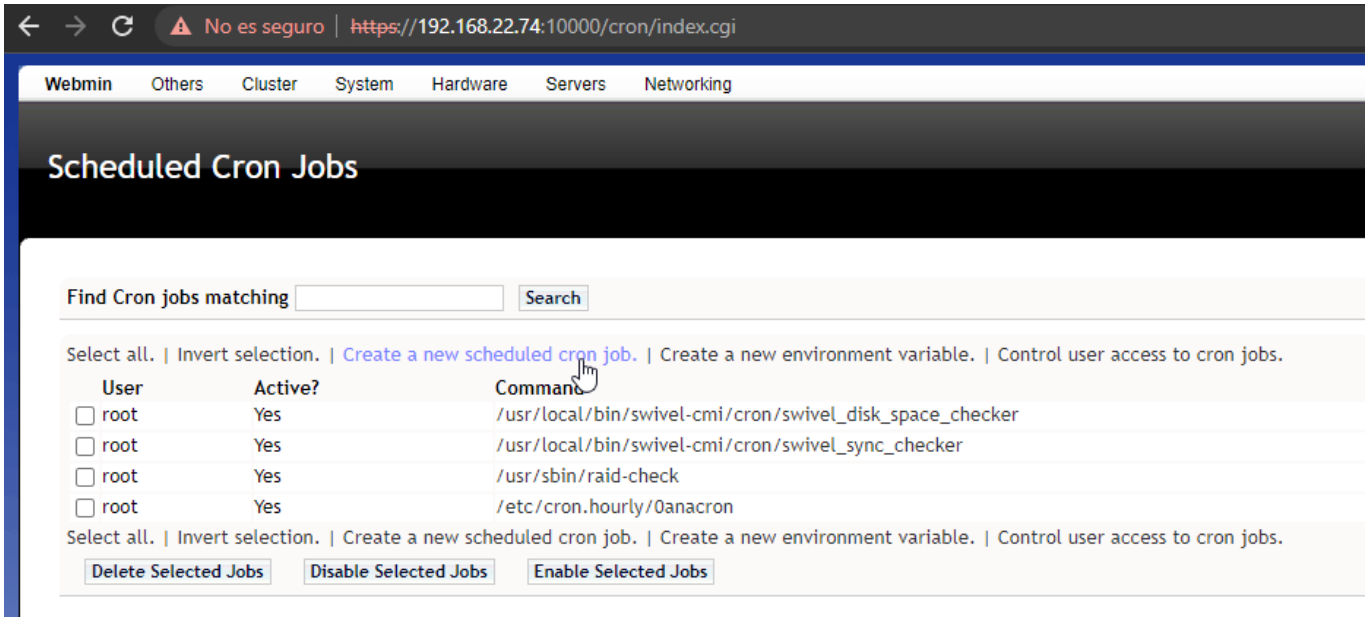
1. Enable webmin from CMI

   `3) Appliance → 2) Start or Stop Services → Webmin (5)`

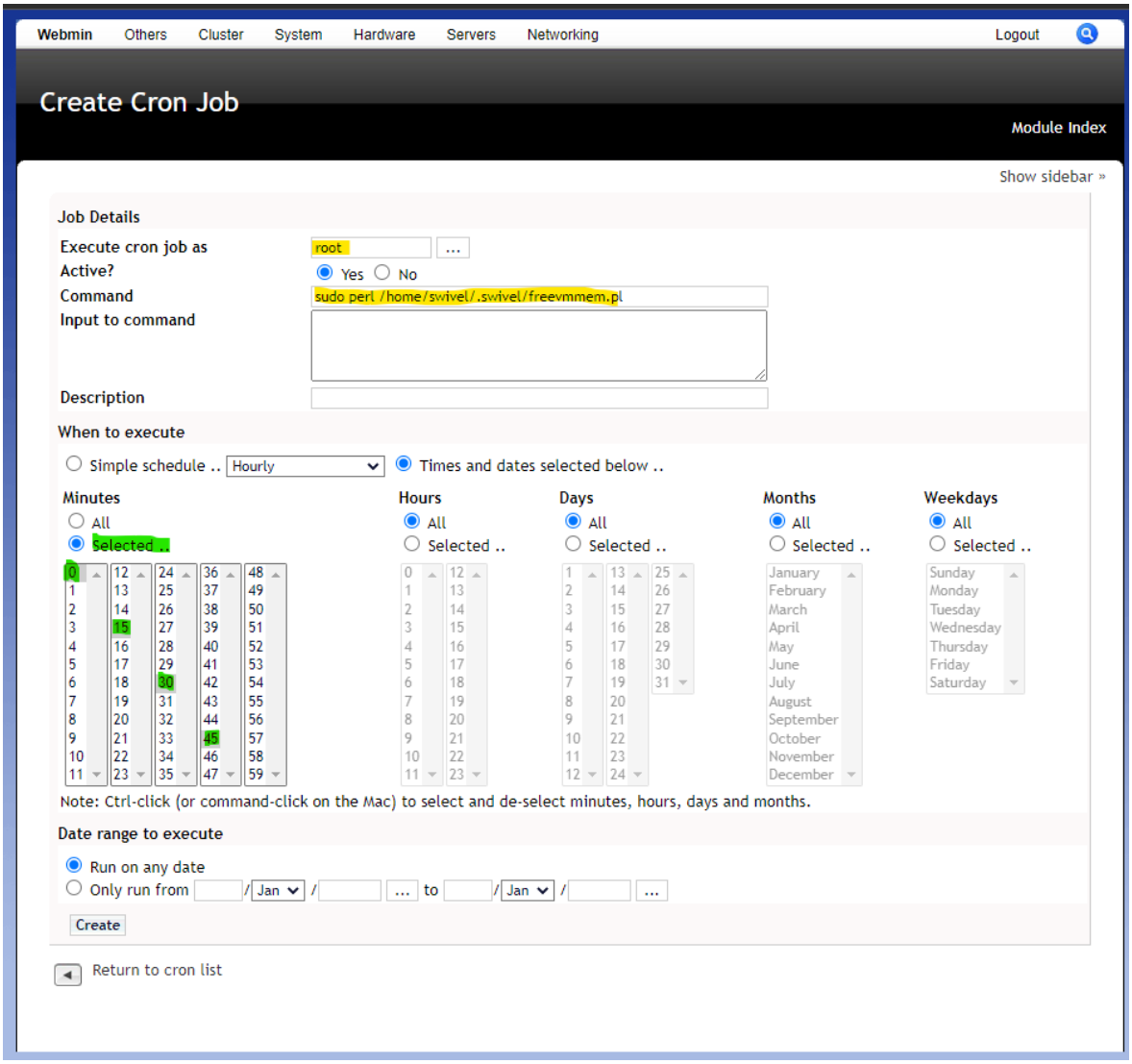
2. Access to Webmin and go to *System → Scheduled Cron Jobs*



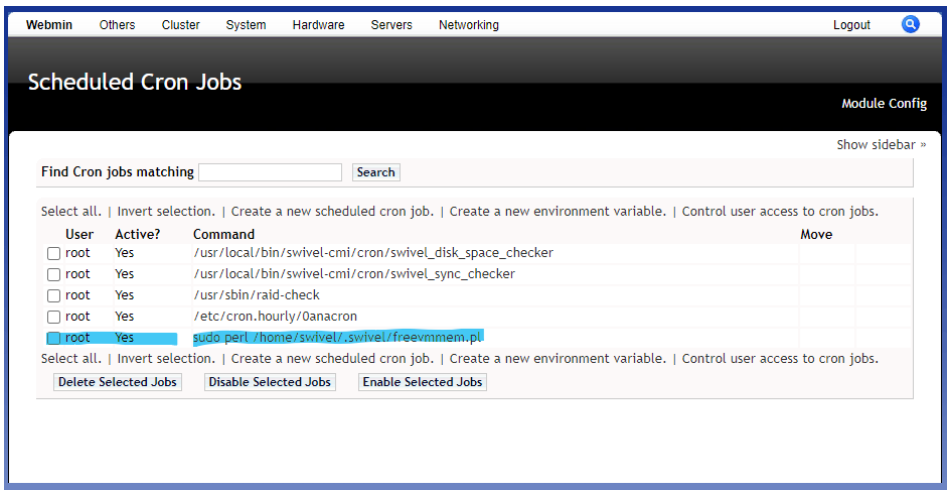
3. *Create a new scheduled cron job*



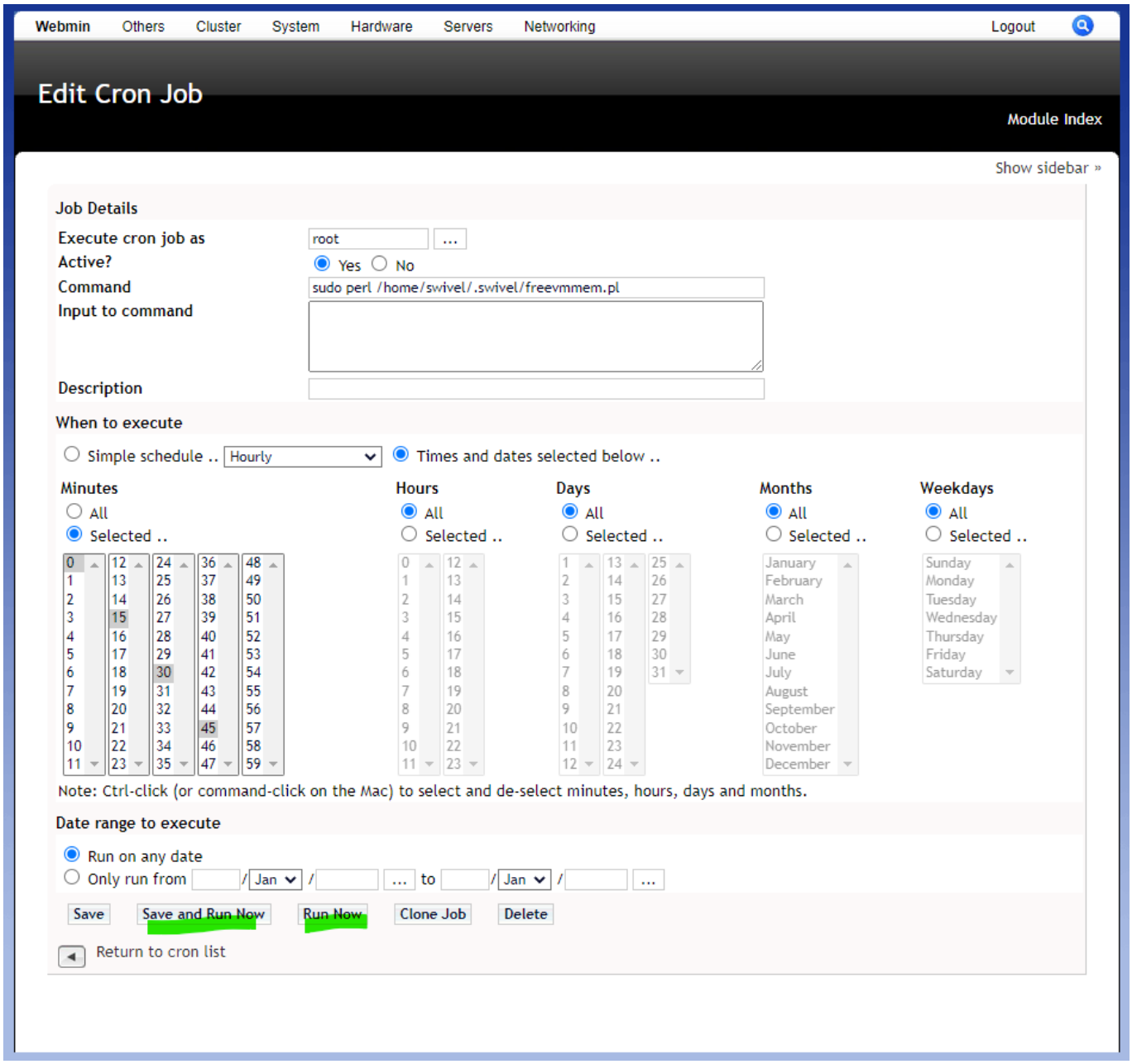
4. Complete the Cront Job with:
    a. Execute as (root)
    b. Command (sudo perl /home/swivel/.swivel/freevmmem.pl)
    c. When to execute (this depends on necessary), this example shows all months, days, hours and minutes 0, 15, 30 and 45 (which means: *every 15 minutes*)



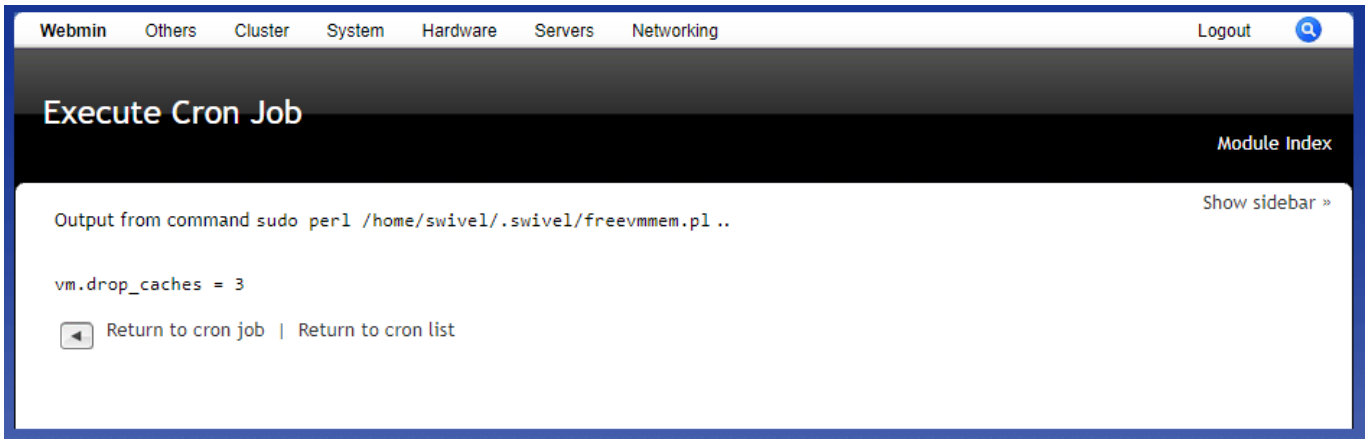
5. Below, press Create to create this con job, which will be listed in the Scheduled Cron Jobs



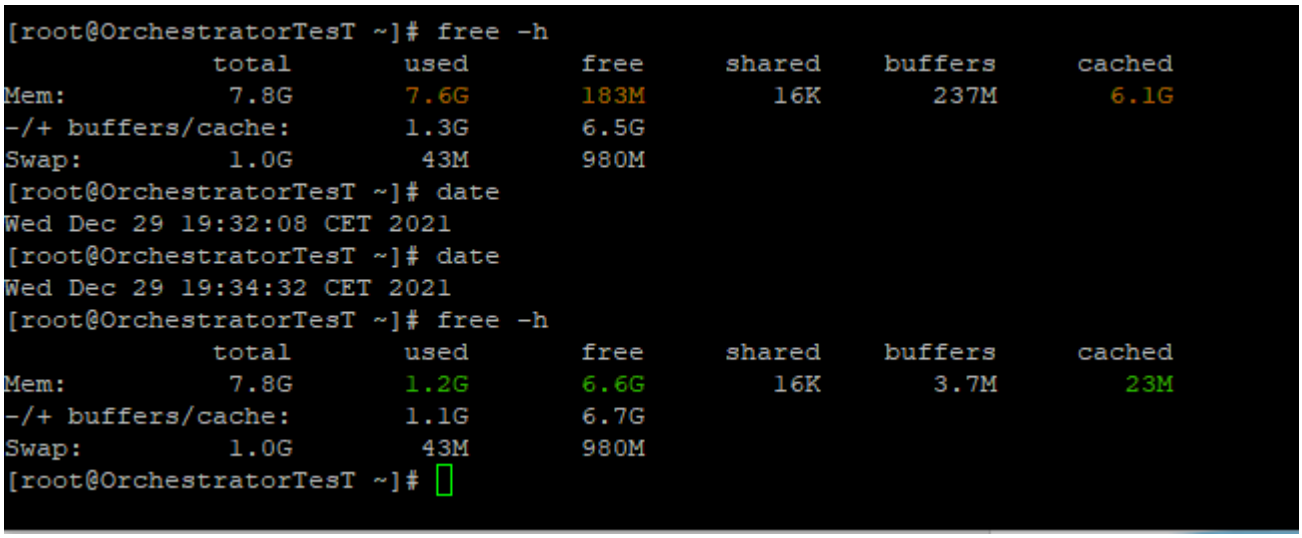
6. Clicking on it from the list, it is possible to Edit in any moment. Also can be Run Now or Save and Run Now to see if everything is fine:



The output for the execution should be like the following image:



The result of the execution (either programmatically, from Webmin or OS command line) can be shown below (example):



More than 6GB of cached memory free, 6.4 GB RAM free.