Nimbus: monitoring

by various contributors
(see last slide)

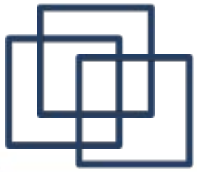
Request for contributors

Hi and thanks for your contribution!

If you have modified something, I'm kindly asking you to
- put your name in the last slide
- write a quick note the next slide concerning what you have modified/added/deleted (since googledocs does not provide this feature, it only says who has modified the presentation and how many slides have been modified)

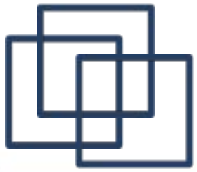
Massimo Canonico (mex@di.unipmn.it)

If you want to became an editor, just send me an email!



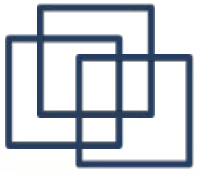
Revision history

14/12/11 - M. Canonico -
14/12/11 - M. Canonico - uploaded on GD and first draft



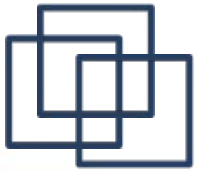
ToDO list

- Fix layout
- Make it less FutureGrid centric
- Mark slides "only for FutureGrid users" when it is necessary
- Improve exercise part



Exercise

- Installation and configuration of monitoring tool in 3 working spaces by using
  - **Nimbus** as cloud platform
  - **Debian-5.0.10-amd64** image as monitoring server
  - **joomlaDebian** images as node to monitor (or any image with a web server)
  - **Munin** as monitoring tool

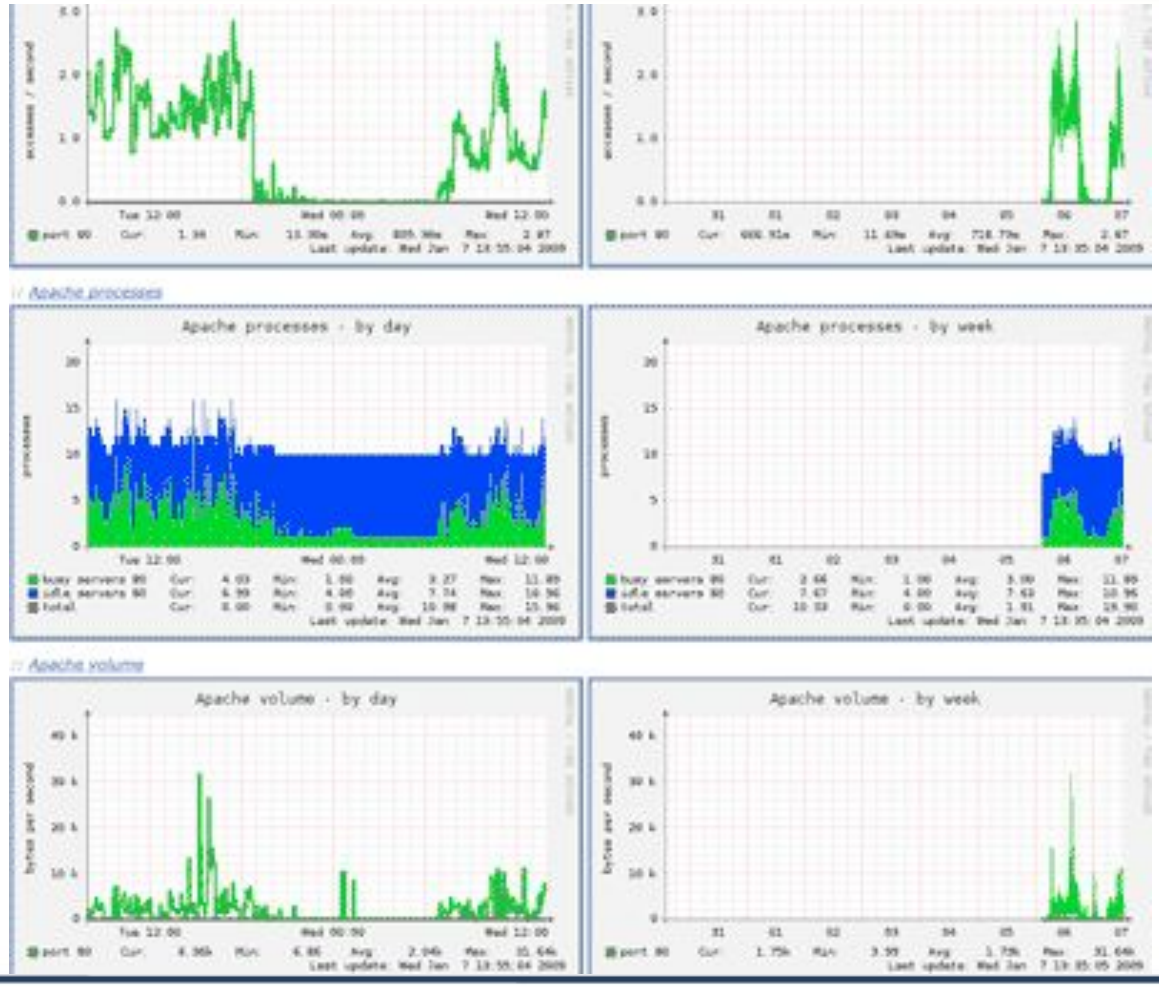


Munin

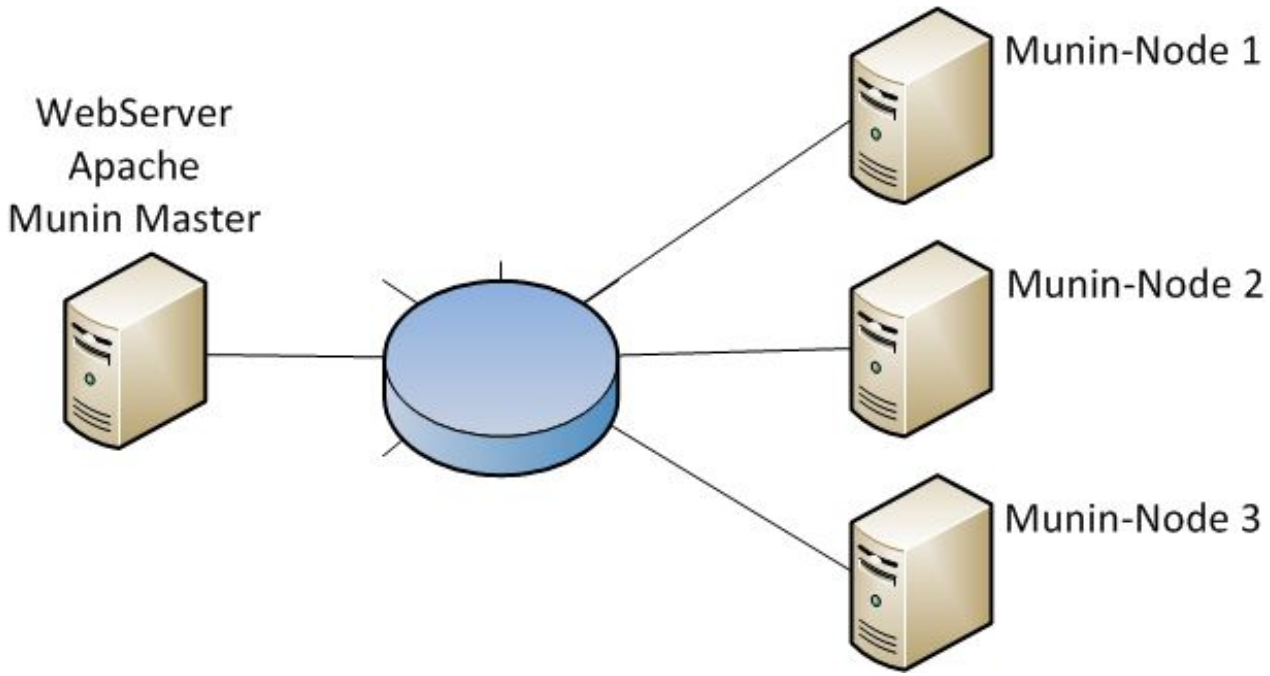
- Munin is a monitoring tool written in **Perl** started by Jimmy Olsen late 2003, based on the excellent RRD tool by Tobi Oetiker. Even if the development has slowed down since 2005, Munin is a stable tool; it is also very widely used, thanks to its **very easy setup**.

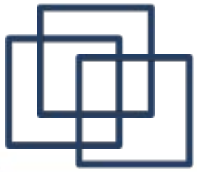
- It consists of ***munin-node***, a daemon you will install on every server you want to monitor and which will gather the data, and ***munin***, which you will install on your monitoring server and which will connect at a regular basis to every node to retrieve it. Munin will then use the data to generate the corresponding **graphs** and **HTML pages**.

Munin


Apache processes
Apache processes - by day
Apache processes - by week
Apache volume
Apache volume - by day
Apache volume - by week

Munin architecture



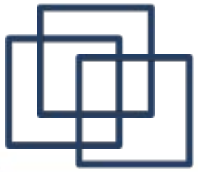


Preparing the instances

- We have three machines: two servers (**munin-nodes**) we want to monitor and a server (**munin-master**) which will monitor them.

- Start 3 workspaces
  - 1 from Debian-5.0.10-amd64.gz image (we'll call it ServerX)
  - 2 from joomlaDebian image (we'll call them ServerA and ServerB)

- Take note of their IP addresses and hostnames
  - in the following slides we consider
    - serverA, IP: aaa.aaa.aaa.aaa (first munin-none)
    - serverB, IP: bbb.bbb.bbb.bbb (second munin-node)
    - serverX, IP: xxx.xxx.xxx.xxx (munin-master)



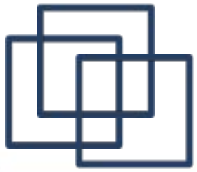
Munin Master: Installation and configuration

- On the monitoring server (serverX), we install munin:
  - sudo apt-get install munin

- We need to tell munin that we want it to monitor serverA and serverB. Munin's configuration file (munin.conf) is usually to be found in /etc/munin/
  - sudo vi /etc/munin/munin.conf

- At the end of the file, add the following:

```
[Domain;serverA]
        address aaa.aaa.aaa.aaa
        use_node_name yes
[Domain;serverB]
        address bbb.bbb.bbb.bbb
        use_node_name yes
```



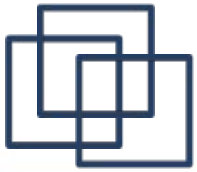
Munin master: installation and configuration (cont)

- serverA/B should be the name of your machines.
  - **Domain** is the "domain" of your machine; in fact it is more a **group name**, used to sort your servers. You can choose to sort by location (server1.london...), by role (server1.apache), or whatever you feel is relevant.

- Munin is a Perl script **run every 5 minutes** by cron. The cronjob should have been set automatically during the installation.
  - Therefore, you don't have to restart it; just wait 5 minutes.

Munin Master: Installation and configuration (cont)

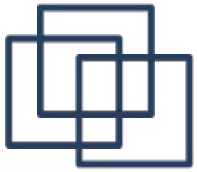
- Munin files for web pages should already be available in /var/www/munin by now. Make them available by installing an HTTP server; **lighttpd** would do the job here.
  - sudo apt-get install lighttpd
    - (it might require to run "apt-get update" before)
- If /var/www/munin does not exist, let's create a symbolic link
  - $ cd /var/www
  - $ ln -s /var/cache/munin/www munin



Installation and configuration Munin-node on ServerA and ServerB

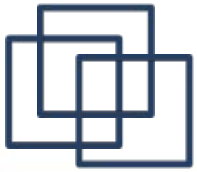
- On the servers we want to monitor, we need to install munin-node.
  - sudo apt-get install munin-node

- By default, only serverA itself will be allowed to connect to this node to retrieve the data; **we need to explicitly allow serverX to connect to it**; this is done at the end of the configuration file of munin-node (usually in /etc/munin/munin-node.conf).
  - sudo vi /etc/munin/munin-node.conf



Installation and configuration Munin-node on ServerA and ServerB (cont.)

- You will find the following line:
  - allow ^127\.0\.0\.1$
- Below, allow the IP address of serverX to connect (the ^ and $ at the beginning and at the end are important):
  - allow ^xxx\.xxx\.xxx\.xxx$
- As munin-node runs as a daemon, you need to restart it to make the changes active.
  - sudo /etc/init.d/munin-node restart
- You can then access the monitoring via your favorite browser with the address **http://xxx.xxx.xxx.xxx/munin/**.



Installation and configuration Munin-node on ServerA and ServerB (cont.)

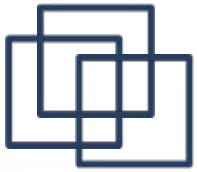
- Increase/decrease usage of
  - CPU, I/O, Disk
    - create a script (many examples available on the Web)
    - use stress project (available here)
  - Network
    - start a scp from/to one of your virtual machines
    - wget of a huge file on Internet

Munin by command line

- Information concerning cpu/network/memory/disk can be retrieved by command line
  - nc/telnet xxx.xxx.xxx.xxx 4949
  - fetch cpu
  - fetch memory
  - ...
- If the firewall on Munin Master does not allow connection to Port 4949, login into Munim Master and use
  - nc/telnet localhost 4949

Video demo

- Video demo of this exercise available here



Bibliography

- These slides are based on this tutorial by Yann Hamon