Lab 2 Monitoring

Grebennikov Sergey, Saad Saif,
Burikova Svetlana, Sharafitdinova Iuliia

February 3, 2018

Task 1 - Practical part

1. Ubuntu 14.04.1 was installed on the Hyper-V virtual machine 1 (VM1). VM1 uses two network adapters for private and public addresses correspondingly.

2. Apache 2.4 and MySQL were installed on VM1.

3. WordPress and PHPMailer 4.7.4 were installed on VM1. A web page with heavy content was created (posts with uploaded videos, images and attachment files). Figure 1
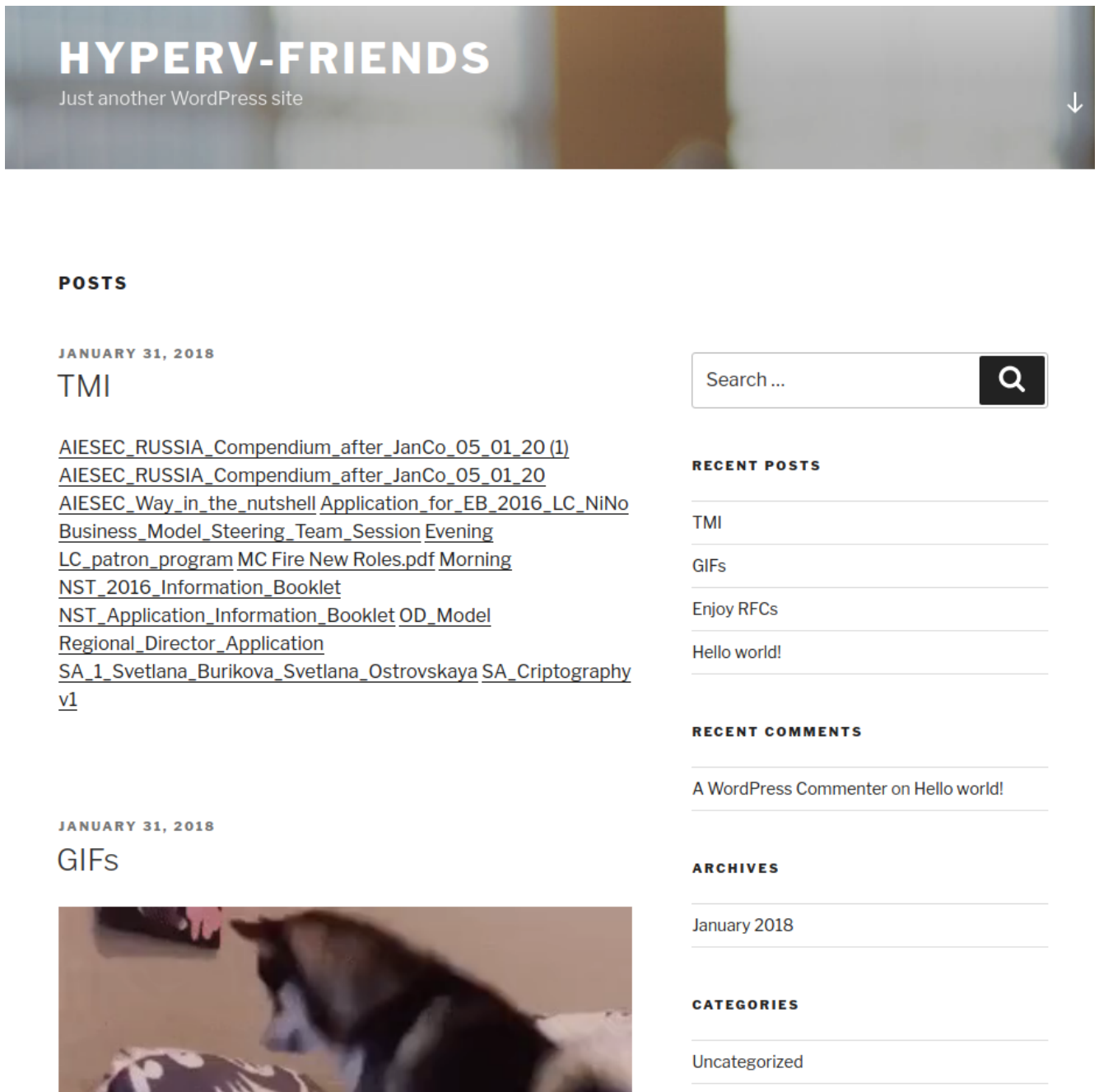

Figure 1: The web page

4. Zabbix agent was installed on VM1, and Zabbix server was installed on a separate virtual machine 2 (VM2). VM2 also uses two network adapters for private and public addresses correspondingly.

5. There were chosen the following status (Figure 2) and trends (Figure 3) metrics:

- Status
  - HTTP service is running. This indicator reflects the current state of the web service.
  - System uptime. It is a measure of the time our web server has been working and available.
  - ICMP response. It uses to test the reachability of a host on the network.
  - CPU load. It shows how much the processor is loaded with different processes.
  - Memory usage. It shows how much memory is allocated to processes.
  - Disk space usage. It shows how full the disk is.
- Trends
  - Apache web server requests
    * Request per sec
    * Request AVG (Total/Uptime)
  - Apache web server connections
    * IDLE workers
    * Busy workers
  - Apache web server traffic
    * KBytes transferred per second
    * Bytes per request
  - MySQL operations
    * MySQL select operation per second
    * MySQL begin operation per second
    * MySQL delete operation per second
    * MySQL insert operation per second
    * MySQL rollback operation per second
    * MySQL update operation per second
  - MySQL bandwidth
    * MySQL bytes received per second
    * MySQL bytes sent per second
    * MySQL queries per second
    * MySQL slow queries
    * MySQL status
    * MySQL uptime
  - Network traffic on eth0 (the private network)
    * Incoming network traffic on eth0
    * Outgoing network traffic on eth0
  - Network traffic on eth1 (the public network)
    * Incoming network traffic on eth1
    * Outgoing network traffic on eht1

Notification system was deployed on Zabbix server (Figure 5). When a critical event occurs (for example, CPU usage or memory usage is too high, a small amount of free HDD space, *etc*), then a trigger is triggered, which notifies the administrator via e-mail about problems.
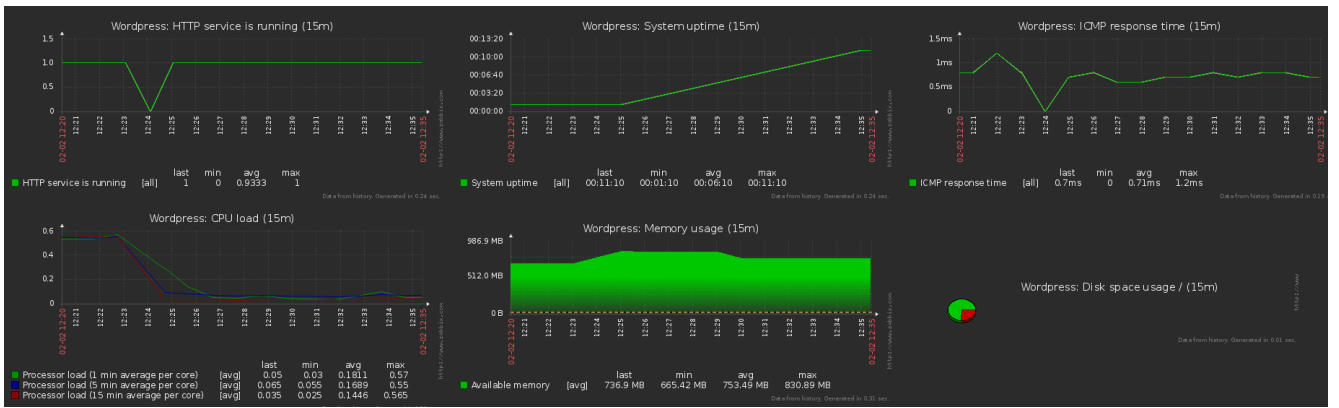
Figure 2: The status metrics

Figure 3: The trends metrics

6. The yandex.tank utility was used for testing purposes. This load-testing tool generated 100.000 requests per 1 minute to the web server. Figure 4

7. The test result might be shown on Figure 5 Figure 6.

Task 2 - Theoretical part

Question 1

How do metrics values match the load generated?

Answer

Figure 5 and Figure 6 shows static metrics that illustrate current state of the component.

Number of running processes can tell us how much processes out of whole amount are running for current time.

Network traffic metric also helps us to monitor the traffic activity.

Figure 4: The load-testing tool yandex.tank

Figure 5: The Zabbix server screen

Figure 6: The Zabbix server screen

Memory usage also help us to monitor memory space available for operations.

CPU load shows how fast CPU is able to run tasks.

Question 2

Evaluate the representativeness of your metrics set: how does it reflect the actual state of affairs?
    Answer

We estimate metrics set as acceptable if it gives us necessary information about system's performance. Presented set of metrics gives us quite exhaustive demonstration of current state and trends. Moreover metrics that notify about resource consumption have benchmarks (figure 4, orange lines). If system cross benchmark by consuming some resource, notification goes to administrator.

Question 3

What is the highload? Can you consume that you've just deployed the highload system? Why?
    Answer

From the engineering perspective, highload is a lack of resources causes performance degradation. Serving 10K+ users simultaneously becomes hard, infrastructure cannot consume incoming streams of data and need in horizontal scaling arise.

Highly loaded systems are web sites with a very large audience, and as a consequence with a heavy load that requires an optimized server part of the website. A qualitative characteristic for a highly loaded system is the capacity of this system, it describes the amount of work that a system must be able to perform per unit of time.

Examples of highly loaded systems can serve as websites such as Google, Facebook, Vkontakte, Flickr, Amazon, YouTube, Wikipedia, LiveJournal, Twitter, MySpace and many others. Also, for high-loaded

systems, you can include browser-based multiplayer games.

Our system can not be named as highload system as it is able to serve only 20% of all requests $(20,000[resp]/100,000[req])$ (Figure 4).