# Практика: DOB Модул 6

## Подготовка на средата

Създаваме папка на хоста и влизаме в нея:

**mkdir DOB-Nagios**

**cd DOB-Nagios**

Разархивираме изтегления файл тук и изпълняваме командата:

**cp Vagrantfile-1 Vagrantfile**

За да подготвим средата за първата част от упражненията.

Можем да разгледаме съдържанието му:

**cat Vagrantfile**

Същинското вдигане на средата започва с изпълняването на командата:

**vagrant up**

След като се стартира машината, изпълняваме:

**vagrant ssh master**

## Част 1

#### Инсталация на Nagios

Преди да стартираме процеса на инсталация, тъй като за целите на текущото упражнение ще използваме като източник пакетното хранилище на дистрибуцията, трябва да се уверим, че **EPEL** е наличен. След това можем да изключим **SELinux**, като редактираме файла:

**sudo vi /etc/sysconfig/selinux**

и сменим стойността

**SELINUX=enforcing**

на

**SELINUX=disabled** или **permissive**

Сега рестартираме с **reboot**.

Нека сега пристъпим към изпълнение на последователността от команди:

**sudo yum install -y nagios**

**sudo systemctl enable httpd**

**sudo systemctl start httpd**

**sudo systemctl status httpd**

**sudo systemctl enable nagios**

**sudo systemctl start nagios**

**sudo systemctl status nagios**

Преди да продължим нататък трябва да настроим защитната стена:

**sudo firewall-cmd --add-service={http,https} --permanent**

**sudo firewall-cmd --reload**

Сега можем да стартираме браузъра на нашата станция и да въведем следния адрес:

[**http://localhost:8000**](http://localhost:8000)

След това можем да сменим адреса на:

[**http://localhost:8000/nagios**](http://localhost:8000/nagios)

От заредената страница разбираме, че преди да можем да работим с **Nagios**, трябва да дефинираме потребител и парола. Това става като изпълним:

**su -**

**htpasswd /etc/nagios/passwd nagiosadmin**

Паролата е **vagrant**. След това въвеждаме примерно **Password1**

Нека сега да излезем от **root** сесията и да рестартираме **httpd**

**exit**

**sudo systemctl restart httpd**

Ако сега се върнем отново браузъра, би трябвало като използваме двойката потребител и парола да можем да влезем в системата.

Нека да разгледаме секциите **Tactical Overview**, **Map (Legacy)**, **Hosts** и **Services**

Явно има проблем с нашата инсталация. Всъщност проблемът идва от липсата на плъгини.

Всъщност можем да инсталираме само част от плъгините, примерно:

**sudo yum install nagios-plugins-{ping,disk,users,procs,load,swap,ssh,http}**

Но ние ще инсталираме всички. Нека да се върнем в конзолата и да ги инсталираме:

**sudo yum install -y nagios-plugins-all**

Сега трябва да рестартираме **nagios**:

**sudo systemctl restart nagios**

**sudo systemctl status nagios**

Нека се върнем отново в браузъра и да проверим ефекта от последната корекция.

Нека сега да се върнем в командния ред и да разгледаме конфигурационните папки и файлове.

След това нека прегледаме папката с плъгини.

Нека да изследваме как се работи с някой от тях в командния ред. Нека това да е **check\_ping**

**su -**

**cd /usr/lib64/nagios/plugins/**

**./check\_ping -h**

**./check\_ping -H 8.8.8.8 -w 2,20% -c 5,50%**

Можем да изпробваме няколко комбинации и/или други плъгини.

Нека се върнем отново в браузъра и да обърнем внимание на **HTTP** проверката. Можем да заключим, че има известен проблем.

Можем да сложим един семпъл **index.html** файл в папката **/var/www/html** със следното примерно съдържание:

**<h1>This is a demo site</h1>**

Сега можем да се върнем отново в браузъра и да наблюдаваме какво се случва с предупреждението.

Ако не искаме да чакаме до следващата проверка, можем да кликнем върху **HTTP** проверката. След това да изпълним командата (от менюто вдясно) - **Re-schedule the next check of this service**. Потвърждаваме с **Commit** и после с **Done**.

Сега можем отново да проверим статуса на проверката.

#### Смяна на външния вид на Nagios

Аналогично на възможностите за добавяне на модули (плъгини), можем да подменяме интерфейса и/или неговата тема (външен вид).

Нека да изтеглим нова тема. За целта трябва да изпълним командата:

**wget http://www.be-root.com/downloads/nagios/vautour/vautour\_style.zip -O vatutor\_style.zip**

След това трябва да подменим съдържанието на папка **/usr/share/nagios/html** с файловете, които се намират в изтегления архив:

**unzip vatutor\_style.zip -d /usr/share/nagios/html**

Сега можем да се върнем в отворения браузър и да обновим заредената страница.

## Част 2

#### Работа с Nagios

Ако все още сме в сесия към машината от предходното упражнение, трябва да излезем и да изпълним следната поредица от команди. Излизаме от **ssh** сесията на **master**:

**exit**

Сега подменяме **Vagrantfile**:

**cp Vagrantfile-2 Vagrantfile**

И изпълняваме:

**vagrant up**

Сега вече трябва да имаме две машини - **master** и **node1**.

Нека да влезем на новата машина и да инсталираме и конфигурираме уеб сървър:

**vagrant ssh node1**

**sudo yum install -y httpd**

**sudo systemctl enable httpd**

**sudo systemctl start httpd**

**sudo firewall-cmd --add-service={http,https} --permanent**

**sudo firewall-cmd --reload**

Сега можем да добавим семпъл тестови **index.html** файл в папката **/var/www/html** със съдържание:

**<h1>Remote host: Demo site.</h1>**

Сега можем да стартираме браузъра на нашата станция и да въведем следния адрес:

[**http://localhost:8001**](http://localhost:8001)

За да тестваме дали всичко работи.

Нека сега да излезем от тази сесия и да се върнем на първия хост:

**exit**

**vagrant ssh master**

Сега можем да пристъпим към добавяне на новия хост в конфигурациите на **Nagios**.

Нека да изпълним следната последователност от команди:

**su -**

**cd /etc/nagios**

**ls -al**

**mkdir -p objects/dob-files**

**cd objects/dob-files**

Създаваме файл **host-node1.cfg** със следното съдържание:

**define host {**

**use linux-server**

**host\_name node1.sulab.local**

**alias Slave Host Node 1**

**address 192.168.82.101**

**icon\_image linux40.png**

**statusmap\_image linux40.png**

**}**

Създаваме шаблон на отдалечена услуга **remote-service-template.cfg**:

**define service {**

**name remote-service**

**use generic-service**

**max\_check\_attempts 5**

**normal\_check\_interval 2**

**retry\_check\_interval 1**

**register 0**

**}**

Създаваме файл **service-http-node1.cfg** със следното съдържание:

**define service {**

**use remote-service**

**host\_name node1.sulab.local**

**service\_description HTTP**

**check\_command check\_http!$HOSTADDRESS$**

**notifications\_enabled 0**

**}**

Правим следните промени в конфигурацията на **Nagios** - файла **/etc/nagios/nagios.cfg**

**cfg\_dir=/etc/nagios/objects/dob-files**

След това проверяваме коректността на конфигурацията с командата:

**nagios -v nagios.cfg**

Ако всичко е наред, можем да рестартираме услуга с:

**systemctl restart nagios**

Можем да се върнем в браузъра в конзолата на **Nagios** и да изследваме промените.

Сега нека се върнем в конзолата и да модифицираме и дефинираме група от хостове.

Нека да редактираме файла **/etc/nagios/objects/localhost.cfg** и да добавим нашия втори хост в групата **linux-servers**. Секцията трябва да придобие вида:

**define hostgroup {**

**hostgroup\_name linux-servers**

**alias Linux Servers**

**members localhost,node1.sulab.local**

**}**

Сега нека да направим наша група. Добавяме файл **/etc/nagios/objects/dob-files/group-dob-hosts.cfg** със следното съдържание:

**define hostgroup {**

**hostgroup\_name dob-hosts**

**alias DOB Demo Servers**

**members node1.sulab.local**

**}**

Нека да проверим конфигурацията и да рестартираме **Nagios**:

**nagios -v nagios.cfg**

**systemctl restart nagios**

Можем да се върнем в браузъра в конзолата на **Nagios** и да изследваме промените.

Сега нека се върнем в конзолата и да добавим две групи от услуги.

Сега нека да направим първата наша група. Добавяме файл **/etc/nagios/objects/dob-files/sgroup-http-services.cfg** със следното съдържание:

**define servicegroup {**

**servicegroup\_name sgroup-http-services**

**alias HTTP Services**

**members localhost, HTTP, node1.sulab.local, HTTP**

**}**

Сега нека да направим втората наша група. Добавяме файл **/etc/nagios/objects/dob-files/sgroup-infra-services.cfg** със следното съдържание:

**define servicegroup {**

**servicegroup\_name sgroup-infra-services**

**alias Infrastructure Services**

**members localhost, Current Load**

**}**

Нека да проверим конфигурацията и да рестартираме **Nagios**:

**nagios -v nagios.cfg**

**systemctl restart nagios**

Можем да се върнем в браузъра в конзолата на **Nagios** и да изследваме промените.

## Част 3

#### Инсталиране на NRPE на подчинен хост

Ако все още сме в сесия към машината от предходното упражнение, трябва да излезем и да изпълним следната поредица от команди. Излизаме от **ssh** сесията:

**exit**

Сега подменяме **Vagrantfile**:

**cp Vagrantfile-3 Vagrantfile**

И изпълняваме:

**vagrant up**

Сега вече трябва да имаме три машини - **master**, **node1** и **node2**.

Нека да влезем в новата машина.

**vagrant ssh node2**

Нека изпълним следната команда, за да инсталираме **NRPE**:

**sudo yum install -y nrpe nrpe-selinux nagios-plugins-nrpe nagios-plugins-all**

След това можем да изключим **SELinux**, като редактираме файла:

**sudo vi /etc/sysconfig/selinux**

и сменим стойността

**SELINUX=enforcing**

на

**SELINUX=disabled** или **permissive**

Сега рестартираме с **reboot**.

Сега можем да редактираме файла **/etc/nagios/nrpe.cfg** като коригираме следните редове:

**…**

**allowed\_hosts=127.0.0.1,::1,192.168.82.0/24**

**…**

**dont\_blame\_nrpe=1**

**…**

**allow\_bash\_command\_substitution=1**

**…**

После разрешаваме съответния порт в защитната стена:

**sudo firewall-cmd --add-port=5666/tcp --permanent**

**sudo firewall-cmd --reload**

След това стартираме услугата:

**sudo systemctl enable nrpe**

**sudo systemctl start nrpe**

**sudo systemctl status nrpe**

И проверяваме дали можем локално да осъществим връзка:

**sudo /usr/lib64/nagios/plugins/check\_nrpe -H localhost**

**sudo /usr/lib64/nagios/plugins/check\_nrpe -H 192.168.82.102**

Сега можем да превключим към главния хост и да пробваме връзка от него:

**exit**

**vagrant ssh master**

За целта първо трябва да добавим още нещо:

**sudo yum install -y nagios-plugins-nrpe**

После да изпълним:

**sudo /usr/lib64/nagios/plugins/check\_nrpe -H 192.168.82.102**

Сега след като всичко е готово, остават само още няколко довършителни действия. За наше улеснение, можем да ги изпълним като **root**:

**su -**

Във файла **/etc/nagios/objects/commands.cfg** добавяме:

**define command {**

**command\_name check\_nrpe**

**command\_line $USER1$/check\_nrpe -H $HOSTADDRESS$ -c $ARG1$**

**}**

Сега ще добавим новия хост като файл **host-node2.cfg** в папката **/etc/nagios/objects/dob-files**:

**define host {**

**use linux-server**

**host\_name node2.sulab.local**

**alias Slave Host Node 2**

**address 192.168.82.102**

**icon\_image linux40.png**

**statusmap\_image linux40.png**

**}**

След това услуга **service-cpu-node2.cfg** папката **/etc/nagios/objects/dob-files**

**define service {**

**use remote-service**

**host\_name node2.sulab.local**

**service\_description CPU Load**

**check\_command check\_nrpe!check\_load**

**}**

Преди за завършим, нека да добавим новия хост и услуга в двете групи от хостове и в едната група услуги.

Редактираме файла **/etc/nagios/objects/localhost.cfg**:

**define hostgroup {**

**hostgroup\_name linux-servers**

**alias Linux Servers**

**members localhost,node1.sulab.local,node2.sulab.local**

**}**

После и файла **/etc/nagios/objects/dob-files/group-dob-hosts.cfg**:

**define hostgroup {**

**hostgroup\_name dob-hosts**

**alias DOB Demo Servers**

**members node1.sulab.local,node2.sulab.local**

**}**

И най-накрая файла с групата от услуги **/etc/nagios/objects/dob-files/sgroup-infra-services.cfg**:

**define servicegroup {**

**servicegroup\_name sgroup-infra-services**

**alias Infrastructure Services**

**members localhost, Current Load, node2.sulab.local, CPU Load**

**}**

Нека да проверим конфигурацията и да рестартираме **Nagios**:

**nagios -v /etc/nagios/nagios.cfg**

**systemctl restart nagios**

Можем да се върнем в браузъра в конзолата на **Nagios** и да изследваме промените.

#### Наблюдение на MariaDB

Инсталираме и конфигурираме базата на последния сървър:

**sudo yum install -y mariadb mariadb-server**

**sudo systemctl start mariadb**

**sudo systemctl enable mariadb**

Влизаме в базата за да направим потребител:

**mysql -u root -p**

**create user nagios@192.168.82.100 identified by 'Password1';**

**quit**

Разрешаваме порта в защитната стена:

**sudo firewall-cmd --add-port=3306/tcp --permanent**

**sudo firewall-cmd --reload**

Връщаме се на основния хост:

**exit**

**vagrant ssh master**

**su -**

За да проверим дали имаме връзка и изпълняваме:

**/usr/lib64/nagios/plugins/check\_mysql -H 192.168.82.102 -u nagios -p Password1**

Създаваме файл с команди **mysql-commands.cfg** в нашата работна папка **/etc/nagios/objects/dob-files**

**# 'check\_mysql' command definition**

**define command {**

**command\_name check\_mysql**

**command\_line $USER1$/check\_mysql -H '$HOSTADDRESS$'**

**}**

**# 'check\_mysql\_cmdlinecred' command definition**

**define command {**

**command\_name check\_mysql\_cmdlinecred**

**command\_line $USER1$/check\_mysql -H '$HOSTADDRESS$' -u '$ARG1$' -p '$ARG2$'**

**}**

**# 'check\_mysql\_database' command definition**

**define command {**

**command\_name check\_mysql\_database**

**command\_line $USER1$/check\_mysql -d '$ARG3$' -H '$HOSTADDRESS$' -u '$ARG1$' -p '$ARG2$'**

**}**

**# 'check\_mysql\_query' command definition**

**define command {**

**command\_name check\_mysql\_query**

**command\_line $USER1$/check\_mysql\_query -q '$ARG4$' -d '$ARG3$' -H '$HOSTADDRESS$' -u '$ARG1$' -p '$ARG2$'**

**}**

След това дефинираме нова услуга **service-mysql-node2.cfg**:

**define service {**

**use remote-service**

**host\_name node2.sulab.local**

**service\_description MYSQL**

**check\_command check\_mysql\_cmdlinecred!nagios!Password1**

**notification\_interval 2**

**}**

Нека да проверим конфигурацията и да рестартираме **Nagios**:

**nagios -v /etc/nagios/nagios.cfg**

**systemctl restart nagios**

Можем да се върнем в браузъра в конзолата на **Nagios** и да изследваме промените.

#### Наблюдение на Docker

Влизаме на последния хост.

Използвайки скрипта **docker\_setup.sh** инсталираме и конфигурираме **Docker**. Имаме два варианта - да го наберем или да го копираме. Нека изберем втория и от нашата станция да изпълним:

**scp -P 2201 docker\_setup.sh vagrant@127.0.0.1:.**

Където порт **2201** е порта на хоста, към който е пренасочен порт **22** на **node2**. Ако не можем да си спомним съответния порт, то можем да изпълним командата:

**vagrant ssh-config**

След това правим файла изпълним и го стартираме.

След успешната инсталация, правим и следните две промени - редактираме **sudo** правата:

**nrpe ALL=(ALL) NOPASSWD: ALL**

И добавяме **nrpe** в групата **docker**:

**sudo usermod -aG docker nrpe**

След успешното инсталиране и конфигуриране, можем да продължим нататък. Рестартираме **node2**.

Влизаме в последния хост и стартираме контейнер:

**docker container run -d --name dob-con alpine sleep 1d**

Създаваме или копираме файла **check\_docker\_container.sh** в папката **/usr/lib64/nagios/plugins/.** За да спестим време, ще го копираме от хоста. Това би могло да стане като изпълним:

**scp -P 2201 check\_docker\_container.sh vagrant@127.0.0.1:.**

Където трябва да заместим порт **2201** с този, към който е пренасочен порт **22** на **node2**. Правим го изпълним.

Във файла **/etc/nagios/nrpe.cfg** добавяме:

**command[check-docker-container]=/usr/lib64/nagios/plugins/check\_docker\_container.sh $ARG1$**

Рестартираме **Nagios** агента:

**sudo systemctl restart nrpe**

Правим проверка като изпълним:

**/usr/lib64/nagios/plugins/check\_nrpe -H 192.168.82.102 -c check-docker-container -a dob-con**

Излизаме от **node2**.

Сега можем да влезем на master хоста:

**vagrant ssh master**

Тестваме:

**/usr/lib64/nagios/plugins/check\_nrpe -H 192.168.82.102 -c check-docker-container -a dob-con**

Можем да превключим:

**su -**

Добавяме нова команда в **/etc/nagios/objects/commands.cfg**:

**define command {**

**command\_name check\_nrpe\_arg**

**command\_line $USER1$/check\_nrpe -H $HOSTADDRESS$ -c $ARG1$ -a $ARG2$**

**}**

Добавяме нова услуга **service-docker-node2.cfg** в папката **/etc/nagios/objects/dob-files**

**define service {**

**use remote-service**

**host\_name node2.sulab.local**

**service\_description Container Running**

**check\_command check\_nrpe\_arg!check-docker-container!dbo-con**

**}**

Нека да проверим конфигурацията и да рестартираме **Nagios**:

**nagios -v /etc/nagios/nagios.cfg**

**systemctl restart nagios**

Можем да се върнем в браузъра в конзолата на **Nagios** и да изследваме промените.