**Lab 14**

**Laboratory Exercise**

**Part 1: Nagios Local Monitoring**

**LAB EXERCISE**

This lab will cover the configuration of Nagios for local monitoring.

**Time to Complete**

Approximately 30 Minutes

**What You Need**

You will need to have completed the following from Labs 11 to 13:

* Nagios Core software downloaded and compiled.
* Nagios Core software installed and ready.
* Nagios Plugins downloaded and compiled.
* Nagios Plugins installed and configured.
* Apache web server installed and running on Nagios server.
* Nagios web interface can be accessed.
* One Apache web server client container installed and running.
* One Postgres database client container installed and running.

Monitoring Apache Web Container

1. Login to your Nagios Server and execute the command below:

*dockeradm@sddo-vm: cd /usr/local/nagios/etc/objects*

1. Edit the “*localhost.cfg*” file using any of these commands:

*dockeradm@sddo-vm: sudo gedit localhost.cfg*

*dockeradm@sddo-vm: sudo nano localhost.cfg*

*dockeradm@sddo-vm: sudo vim localhost.cfg*

1. Add these lines to the “*localhost.cfg*” file, and save the file:

define host {

use linux-server

host\_name apacheweb

alias apacheweb.localdomain

address 192.168.100.11

}

1. Exit to your Nagios Server system prompt and execute the command/s below.

*dockeradm@sddo-vm: sudo systemctl restart nagios.service*

1. Launch your web browser and access the Nagios web interface. Paste a screenshot of what you have observed as a result of the above configuration.

|  |
| --- |
| <Insert screen capture of results> |

1. Edit the “*localhost.cfg*” file again using any of these commands:

*dockeradm@sddo-vm: sudo gedit localhost.cfg*

*dockeradm@sddo-vm: sudo nano localhost.cfg*

*dockeradm@sddo-vm: sudo vim localhost.cfg*

1. Add these lines to the “*localhost.cfg*” file, and save the file:

define service {

use local-service

host\_name apacheweb

service\_description PING

check\_command check\_ping!100.0,20%!500.0,60%

}

define service {

use local-service

host\_name apacheweb

service\_description SSH

check\_command check\_ssh

notifications\_enabled 0

}

define service {

use local-service

host\_name apacheweb

service\_description HTTP

check\_command check\_http

notifications\_enabled 0

}

define service {

use local-service

host\_name apacheweb

service\_description Current Users

check\_command check\_local\_users!20!50

}

define service {

use local-service

host\_name apacheweb

service\_description Current Load

check\_command check\_local\_load!5.0,4.0,3.0!10.0,6.0,4.0

}

1. Exit to your Nagios Server system prompt and execute the command/s below.

*dockeradm@sddo-vm: sudo systemctl restart nagios.service*

1. Launch your web browser and access the Nagios web interface. Paste a screenshot of what you have observed as a result of the above configuration.

|  |
| --- |
| <Insert screen capture of results> |

1. Edit the “*localhost.cfg*” file using any of these commands:

*dockeradm@sddo-vm: sudo gedit localhost.cfg*

*dockeradm@sddo-vm: sudo nano localhost.cfg*

*dockeradm@sddo-vm: sudo vim localhost.cfg*

1. Amend these lines in the “*localhost.cfg*” file, and save the file:

define hostgroup {

hostgroup\_name linux-servers

members localhost**,apacheweb**

}

1. Exit to your Nagios Server system prompt and execute the command/s below.

*dockeradm@sddo-vm: sudo systemctl restart nagios.service*

1. Launch your web browser and access the Nagios web interface. Paste a screenshot of what you have observed as a result of the above configuration.

|  |
| --- |
| <Insert screen capture of results> |

Monitoring Postgres Database Container

1. Repeat all the above steps for monitoring Apache Web Container, except to change these values:

host\_name postgresdb

alias postgresdb.localdomain

address 192.168.100.12

**Part 2: Nagios Remote Monitoring**

**LAB EXERCISE**

This lab will cover the configuration of Nagios for remote monitoring.

**Time to Complete**

Approximately 30 Minutes

**What You Need**

You will need to have completed the following from Labs 11 to 13:

* Nagios Core software installed and ready.
* Nagios Plugins installed and configured.
* Apache web server installed and running on Nagios server.
* Nagios web interface can be accessed.
* One Apache web server client container installed and running.
* One Postgres database client container installed and running.
* OpenSSH-server package installed on both client containers.
* Password-less SSH login configured.

Monitoring Apache Web Container

1. Login to your Apache Web Container and execute the command below:

*dockeradm@sddo-vm: docker exec --privileged -it apacheweb /bin/bash*

*root@apacheweb: adduser nagios*

1. Login to your Nagios Server and execute the command below:

*dockeradm@sddo-vm: sudo mkdir /home/nagios*

*dockeradm@sddo-vm: sudo chown nagios:nagios /home/nagios*

*dockeradm@sddo-vm: sudo su - nagios*

*$: ssh-keygen -t rsa -b 4096 -C "nagios@localhost"*

*$: ssh-copy-id nagios@192.168.100.11*

*$: exit*

1. Edit the “*commands.cfg*” file using any of these commands:

*dockeradm@sddo-vm: sudo gedit localhost.cfg*

*dockeradm@sddo-vm: sudo nano localhost.cfg*

*dockeradm@sddo-vm: sudo vim localhost.cfg*

1. Add these lines to the “*commands.cfg*” file, and save the file:

define command {

command\_name check\_remote\_load

command\_line /usr/bin/ssh $HOSTADDRESS$ $USER1$/check\_load -w $ARG1$ -c $ARG2$

}

1. Update the *host* and *service* definitions in the “*localhost.cfg*” file, and save the file.
2. Exit to your Nagios Server system prompt and execute the command below:

*dockeradm@sddo-vm: sudo systemctl restart nagios.service*

1. Launch your web browser and access the Nagios web interface. Paste a screenshot of what you have observed as a result of the above configuration.

|  |
| --- |
| <Insert screen capture of results> |

Monitoring Postgres Database Container

1. Repeat all the above steps for monitoring Apache Web Container.

**--End of Lab Exercise --**