**Lab 13**

**Laboratory Exercise**

**Part 1: Monitored Clients**

*This should have been done in Lab 12. This section includes the commands to achieve it.*

**LAB EXERCISE**

This lab will cover the installation of additional client containers.

**Time to Complete**

Approximately 20 Minutes

**What You Need**

You will need to refresh and reuse your Docker container commands from previous lessons.

Client Containers

1. For the first client container, you shall install a Docker container with Apache web server using these commands:

*docker run -d --privileged -h "apacheweb.localdomain" --name apacheweb --add-host "sddo-vm.localdomain:172.20.113.93" --ip "192.168.100.11" -e TZ=SGT -p 8011:80 ubuntu/apache2:latest*

*docker network connect --ip "192.168.100.11" customnetwork apacheweb*

*docker exec --privileged -it apacheweb /bin/bash*

1. Within the container, install SSH server with these commands:

*apt update*

*apt install openssh-server*

1. For the second client container, you shall install a Docker container with Postgres database using similar commands as above.

**Part 2: Install and Configure SSH**

**LAB EXERCISE**

This lab will cover the installation and configuration of SSH.

**Time to Complete**

Approximately 15 Minutes

**What You Need**

You will need to have your client containers up and running.

SSH Installation and Configuration

1. Follow either of these links to install and configure openssh-server package on your client containers:
   1. <https://linuxize.com/post/how-to-enable-ssh-on-ubuntu-18-04/>
   2. <https://linuxhint.com/enable_ssh_server_ubuntu_1804/>
2. Next, follow either of these links to implement password-less SSH login to both your client containers:
   1. <https://websiteforstudents.com/how-to-setup-ssh-key-login-on-ubuntu-linux/>
   2. <https://www.linuxbabe.com/linux-server/setup-passwordless-ssh-login>

Step 1 – Commands at client container (or SSH server):

|  |  |
| --- | --- |
|  | sudo service ssh status |
|  | sudo service ssh restart |

Step 2 – Commands at Nagios server (or SSH client):

|  |  |
| --- | --- |
|  | ssh-keygen -t rsa -b 4096 -C "username@hostname" |
|  | ssh-copy-id username@ssh\_server\_ip |

Step 3 – Commands at client container (or SSH server):

|  |  |
| --- | --- |
|  | sudo nano /etc/ssh/sshd\_config  add/change-> PasswordAuthentication no  add/change-> ChallengeResponseAuthentication no  add/change-> UsePAM no |
|  |  |
|  | sudo service ssh restart |

**Part 3: Nagios Configurations**

**LAB EXERCISE**

This lab will cover the configuration of Nagios monitoring.

**Time to Complete**

Approximately 30 Minutes

**What You Need**

You will need to refer to the lecture notes (PowerPoint file) for Lesson 13.

Nagios Configuration

1. Refer to the slides and configure Hosts, Services, Commands, Contacts, Time Periods, Host group, and Service group.

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: gedit localhost.cfg*

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

*dockeradm@sddo-vm: 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 below:

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

**Summary:**

* 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.
* One Postgres database client container installed.
* OpenSSH-server package installed on both client containers.
* Password-less SSH login configured.

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