Solutions to lab tasks 01-19

1

Done

2

Done. Important to connect the floating IP to the internal network.

3

Installed apache with:

```
    apt-get update
    apt-get install apache2

added PHP and MySQL support with:

    apt-get install libapache2-mod-php php-mysql
    apt-get install mysql-client
```

4

Tested it from manager with `wget -q -O - http://<manager-ip>/`

5

Installed haproxy with:

```
    sudo add-apt-repository -y ppa:vbernat/haproxy-1.8
    apt-get update
    apt-get install haproxy socat
```

configured haproxy by adding the following to /etc/haproxy/haproxy.cfg

```
    frontend bookface
        bind *:80
        mode http
```

```
        default_backend nodes

    backend nodes
        mode http
        balance roundrobin
        server www1 10.10.0.123:80 check
        server www2 10.10.0.134:80 check

    listen stats
        bind *:1936
        stats enable
        stats uri /
        stats hide-version
        stats auth someuser:password
```

Note: had to restart haproxy for it to work. Works with both the private and floating IPs (?)

6

Sent the info. Chose the group name “TnT”

7

(Our best attempt. We don’t feel qualified to give insightful input on this question. We’d love to hear a good answer to this question from you.)

Four tasks/incidents that require coordination between three operations teams; application, database and SAN- and infrastructure, even though it only needs to be handled by one of them:

1. **Application servers hacked.** Only the Application people need to debug it, but the rest of the teams should be aware of the scale of the hack so they can respond if appropriate (it’s their call). The database operators might want to do a search for malicious information in their databases, and the SAN people might want to prepare backups (determine how far they would have to go).
2. **Application server hardware switched out.** Application operators might decide to switch out their hardware for new ones (or their VMs for new ones). Database people should know because they’ll see new hardware suddenly using their databases, which could be suspicious if they weren’t forewarned.
3. **Scaling up/down application servers.** Operators of the application servers should feel free to scale the service up and down as they please, but they have to keep the rest of the teams

up-to-date to avoid scaling issues. The database and the SAN-infrastructure could be overloaded or overpowered.

4. **A database server going down.** It's the database server operators responsability to bring it up again, but the application server operators should be aware, so they don't overload the remaining database server(s).

8

This result tells us the amount of src attributes that point to URLs starting with http. This test is not accurate because the src could be set in JavaScript or other ways we are not aware of. Google gives a result of 0 for example.