1. **TASK- Script**

**Build instructions** : It is asked that writing a scripts about given the randon number from 1-10 order. And each number must be unique. This script will be written by bash.

We use shuf system command for this script. This command is used for random permutation of the input lines to standard output. This command reads either from a file or standard input in bash and randomizes those input lines and displays the output.

**Usage**: the script is like below. It is working scripts home directory.

cd /scripts

chmod 755 random.sh

cat random.sh

#!/bin/bash

shuf -i 1 10;

sh random.sh ---------- output is below.

5

7

10

3

9

1

2

4

8

6

seq 1 10 | sort -R;

**Description**: this command is user for generating unique number between 1 and 10.

**Lmitations**: This scrips is only generate number in 1-10 interval.

Except for the system command we can use below script as alternate;

**Usage**: This script works under own directory.

cd /scripts

chmod 755 random1.sh

cat random1.sh

#!/bin/bash

for (( i=1;i<=10;i++ ))

do echo $RANDOM $i

done|sort -k1 |cut -d" " -f2;

With the for-do loop, it is written random number and 1-10 number in the different columns. After, It is sorted the two columns by random number. In the next stage It is cut first column. So It is obtained random number between 1 and 10 as nonrepeated.

1. **TASK - MONITORING**

CPU spec is like that;

4 times Intel(R) Xeon(R) CPU E7-4830 v4 @ 2.00GHz

https://www.intel.com/content/www/us/en/products/sku/93811/intel-xeon-processor-e74830-v4-35m-cache-2-00-ghz/specifications.html

Some calculations about our tasks and The metrics that must be monitored are below;

1. Server CPU usage: Every request use one thread. Totaly 4 Socket X 28 thread = 112 Thread. 25.000 request / 112 Thread = 223 simultaneous process for one thread.

2.0Ghz = 2.000.000.000 process per second. 2.000.000.000 x 56 (4 socket\*14 core) core = 112.000.000.000 process per second. This is total CPU capacity for processing.

So as it will see the above calculations, CPU usage is very low for this case. But we must monitor CPU usage via standalone or toos like Zabbix, nagios etc.

1. Server Memory Usage : The default HTTP Request Header value is 8190 bytes. 25000 x 8 kb = 150.000 kb= 150MB Memory. We have 64GB memory capacity. There is no bottleneck for the memory.
2. Number of web server process: Every request is a process for operating system
3. Total active connections: There are 223 momentary connection. So we have 223 active connection for this case.
4. Disk need: For ssl offload and proxy operation it is used memory for cache. So it is needed to disk space for only operating system and logs. For this task, 2 TB disk capacity is enough. But we must add to monitoring tool.
5. Network utilization: 25.000 X 8KB = 200000KB = 195MB = 1562Mb/s = 1.53Gb/s

As Conclusion;

For standlone monitoring,

If we use Windows for ssl offload and Proxy application, we must monitor via performans monitoring. We can monitor CPU, RAM, Disk, network, connetion metrics via performance monitoring tool. After we monitor adaquate time interval, we can analyse these data. What is the threshold values for metrics? Which metrics have used as full capacity? We detect all of these. Also we can monitor as live.

For Zabbix or nagios,

If we use Zabbix or nagios (or etc.) monitoring tool. We must add this SSL offload and Proxy server on these. After we added as a host on the monitoring tool, we can describe the items for CPU, RAM, DISK, network and process monitoring options. We can easily monitor this type of centrilized monitoring system.