DOKUMEN PENALAAN SERVER DAN KONFIGURASI SISTEM APLIKASI SEKTOR AWAM

BAGI KETERSEDIAAN, PENGGUNAAN DAN PRESTASI YANG TINGGI

NAMA AGENSI	:	BAHAGIAN PERUNDINGAN ICT
NAMA AGENSI INDUK	:	UNIT PEMODENAN TADBIRAN DAN PERANCANGAN PENGURUSAN MALAYSIA (MAMPU)
TARIKH DOKUMEN	:	18/04/2023
VERSI DOKUMEN	:	2.0

KANDUNGAN

i. Keterangan Dokumen

Dokumen Penalaan Server dan Konfigurasi Sistem Aplikasi Sektor Awam ini disediakan untuk menerangkan mengenai kaedah penalaan dan konfigurasi yang dicadangkan kepada server dan sistem aplikasi bagi menampung bebanan pengguna yang tinggi. Dokumen ini memperincikan cadangan penalaan yang boleh dilaksanakan berdasarkan kepada perkongsian pengalaman melalui projek-projek pengujian prestasi yang telah dilaksanakan sebelum ini.

ii. Semakan dan Pengesahan Dokumen

DOKUMEN DISEDIAKAN OLEH

Nama	Jawatan	Tandatangan	Tarikh
Ts Mohd Hasiady Bin Yasin	Ketua Penolong Pengarah Kanan, Bahagian Perundingan ICT, MAMPU		17/2/2023

DOKUMEN DISEMAK OLEH

Nama	Jawatan	Tandatangan	Tarikh
Ts Ibrahim Bin Abdullah	Ketua Penolong Pengarah, Bahagian Perundingan ICT, MAMPU		17/2/2023
Mohd Syukor Bin Abdul	Penolong Pengarah Kanan, Bahagian Pembangunan Aplikasi, MAMPU		17/2/2023

DOKUMEN DISAHKAN OLEH

Nama	Jawatan	Tandatangan	Tarikh
Hanissull Jalis Binti Md Yusof	Ketua Perunding Pembangunan Sistem, Bahagian Perundingan ICT, MAMPU		17/2/2023

iii. Kawalan Dokumen

No. Versi	Tarikh	Ringkasan Pindaan	Disediakan/ Dipinda oleh
1.0	17/2/2023	Draf pertama	Ts Mohd Hasiady Bin Yasin
2.0	18/4/2023	Pindaan pertama	Ts Mohd Hasiady Bin Yasin

1 PENALAAN BAGI SEMUA SERVER

Semua server perlu dibuat penalaan untuk memastikan server boleh menampung bebanan pengguna yang tinggi dan berupaya melaksanakan proses dalam masa tindakbalas yang sesuai. Penalaan yang dicadangkan adalah bagi sistem pengoperasian sumber terbuka iaitu CentOS, Ubuntu dan RedHat LINUX. Cadangan penalaan adalah bagi spesifikasi server dengan kapasiti 16CPU dan 16GB RAM untuk menampung beban pengguna sehingga 10,000 pengguna serentak (concurrent user).

1.1 Edit limits.conf

> **sudo vi /etc/security/limits.conf**

```
# tambah last row
* hard nofile  5000000
* soft  nofile  5000000
```

atau

```
# tambah last row
* hard nofile  2000000
* soft  nofile  2000000
```

1.2 Edit sysctl.conf

> **sudo vi /etc/sysctl.conf**

```
#tambah last row
vm.max_map_count = 262144
fs.file-max = 5000000
fs.nr_open = 5000000
net.core.somaxconn = 32768
net.ipv4.tcp_notsent_lowat = 16384
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_timestamps = 1
```

atau

```
#tambah last row

vm.max_map_count = 262144

fs.file-max = 20000000

fs.nr_open = 20000000

net.core.somaxconn = 32768

net.core.default_qdisc = fq

net.ipv4.tcp_congestion_control = bbr

net.ipv4.tcp_notsent_lowat = 16384

net.ipv4.tcp_tw_reuse = 1

net.ipv4.tcp_timestamps = 1
```

- **sudo sysctl -p**

  Sepatutnya tiada error dipaparkan

- **sudo systemctl reboot**

  Server akan reboot

2   APACHE MPM PREFORK CONFIGURATION

How to enable Apache MPM Prefork to improve performance and Configure Apache2 - Using Prefork.

How to enable the MPM_Prefork

To check to see if the prefork module is loaded, issue the command:
- **sudo apache2ctl -M | grep prefork**

If you see no results, prefork isn't loaded. Before you do load it, you'll have to first unload the mpm_event module, as they will conflict.

To unload mpm_event, issue the command:
- **sudo a2dismod mpm_event**

Restart Apache with the command:
- **sudo systemctl restart apache2**

Now you can load mpm_prefork with the command:
- **sudo a2enmod mpm_prefork**

Once again, restart Apache with the command:
- **sudo systemctl restart apache2**

How to configure the MPM_Prefork

Edit prefork configuration

➢ **sudo vi /etc/apache2/mods-available/mpm_prefork.conf**

In that file, you'll see the following options:

```
StartServers            5
MinSpareServers         5
MaxSpareServers         10
MaxRequestWorkers       150
MaxConnectionsPerChild  0
```

You might want to bump those numbers up considerably, if you know your server is going to be under a larger load (and you have the RAM to spare), you could alter that configuration like so:

```
StartServers            4
MinSpareServers         3
MaxSpareServers         40
MaxRequestWorkers       200
MaxConnectionsPerChild  10000
```

Make sure to edit those numbers according to what your hardware can handle and your expected load will require. Save and close the file.

Restart Apache with the command:

➢ **sudo systemctl restart apache2**

3   APACHE PHP FPM CONFIGURATION

PHP FPM configuration

pm = dynamic

pm.max_children = 30

pm.start_servers = 15

pm.min_spare_servers = 10

pm.max_spare_servers = 20

Restart Apache with the command:

➤ **sudo systemctl restart apache2**

4   NGINX CONFIGURATION

EDIT NGINX CONFIGURATION

➢ **sudo vi nginx.conf**

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
worker_rlimit_nofile 2000000;
include /etc/nginx/modules-enabled/*.conf;
events {
        worker_connections 3000;
        use epoll;
        multi_accept on;
}

http {
        ## Basic Settings ##
        sendfile on;
        tcp_nopush on;
        types_hash_max_size 2048;
        fastcgi_read_timeout 30;
        proxy_read_timeout 30;
        server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;
```

```
        ## SSL Settings #
        ssl_protocols TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
        ssl_prefer_server_ciphers on;

        ## Logging Settings ##
        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;

        ## Gzip Settings ##
        gzip on;
        gzip_vary on;
        gzip_proxied any;
        gzip_comp_level 6;
        gzip_buffers 16 8k;
        gzip_http_version 1.1;
        gzip_types text/plain text/css application/json application/javascript
text/xml application/xml application/xml+rss text/javascript;

        ## Virtual Host Configs ##
        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;
}
```

Restart Apache with the command:

> **sudo systemctl restart nginx**

EDIT SITES AVAILABLE

> **vi siteavailable**

Default server configuration#
```
server {
        root /var/www/mygovevent/public;
        # Add index.php to the list if you are using PHP
        index index.php index.html;
        client_max_body_size 40M;
        server_name mygovevent.mampu.gov.my;
        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                # try_files $uri $uri/ =404;
              try_files $uri $uri/ /index.php?$query_string;
        }


        # pass PHP scripts to FastCGI server #
        location ~ \.php$ {
                include snippets/fastcgi-php.conf;
                fastcgi_pass unix:/run/php/php8.0-fpm.sock;
                fastcgi_param SCRIPT_FILENAME
$realpath_root$fastcgi_script_name;
                include fastcgi_params;

            # With php-cgi (or other tcp sockets):
            #fastcgi_pass 127.0.0.1:9000;
            #fastcgi_index index.php;
```

```
        }

        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        location ~ /\.ht {
                deny all;
        }


    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate
/etc/letsencrypt/live/mygovevent.mampu.gov.my/fullchain.pem; # managed by
Certbot
    ssl_certificate_key
/etc/letsencrypt/live/mygovevent.mampu.gov.my/privkey.pem; # managed by
Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}


server {
    if ($host = mygovevent.mampu.gov.my) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


        listen 80 default_server;
        listen [::]:80 default_server;


        server_name mygovevent.mampu.gov.my;
    return 404; # managed by Certbot
}
```

5   PHP.INI CONFIGURATION

<u>Edit php.ini configuration</u>

memory_limit = 256M

post_max_size = 40M

upload_max_filesize = 40M

max_file_uploads = 20

5   PHP.INI CONFIGURATION

6   MY.CNF CONFIGURATION

> **Edit my.cnf**

```
 [client-server]
# # include all files from the config directory #
includedir /etc/my.cnf.d


[mysqld]
max_connections = 2000
bind_address=0.0.0.0
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
log-error=/var/log/mysql/mysqld.log
pid-file=/run/mysqld/mysqld.pid


# Performance
# Added by xxx on 11 Feb 2023
innodb_buffer_pool_size = 30G
innodb_buffer_pool_instances = 30
innodb_flush_log_at_trx_commit = 0
innodb_flush_method = O_DIRECT
innodb_log_buffer_size = 8M
innodb_log_file_size = 3G


key_buffer_size = 256M
max_allowed_packet=1026M
table_open_cache=40000
open_files_limit=500000
sort_buffer_size = 1M
```

```
read_buffer_size = 1M
read_rnd_buffer_size = 4M
myisam_sort_buffer_size = 64M


slow_query_log = on
long_query_time = 2
slow-query_log_file = /var/log/mysql/mysql-slow.log


# Added by xxx on 11 Feb 2023
min_examined_row_limit=10000
log_queries_not_using_indexes=1
```

7   POSTGRE SQL CONFIGURATION

➢ **Exit postgresql.conf**

max_connections = 500

shared_buffers = 2GB

work_mem = 1MB

dynamic_shared_memory_type = posix

effective_io_concurrency = 300

max_worker_processes = 8

max_parallel_maintenance_workers = 4

max_parallel_workers_per_gather = 4

max_parallel_workers = 8

wal_buffers = 16MB

max_wal_size = 4GB

min_wal_size = 1GB

effective_cache_size = 6GB

➢ **Edit sysctl.conf**

kernel.shmmax = 16777216

vm.max_map_count = 262144

fs.file-max = 20000000

fs.nr_open = 20000000

net.core.somaxconn = 32768

net.core.default_qdisc = fq

net.ipv4.tcp_congestion_control = bbr

net.ipv4.tcp_notsent_lowat = 16384

net.ipv4.tcp_tw_reuse = 1

net.ipv4.tcp_timestamps = 1