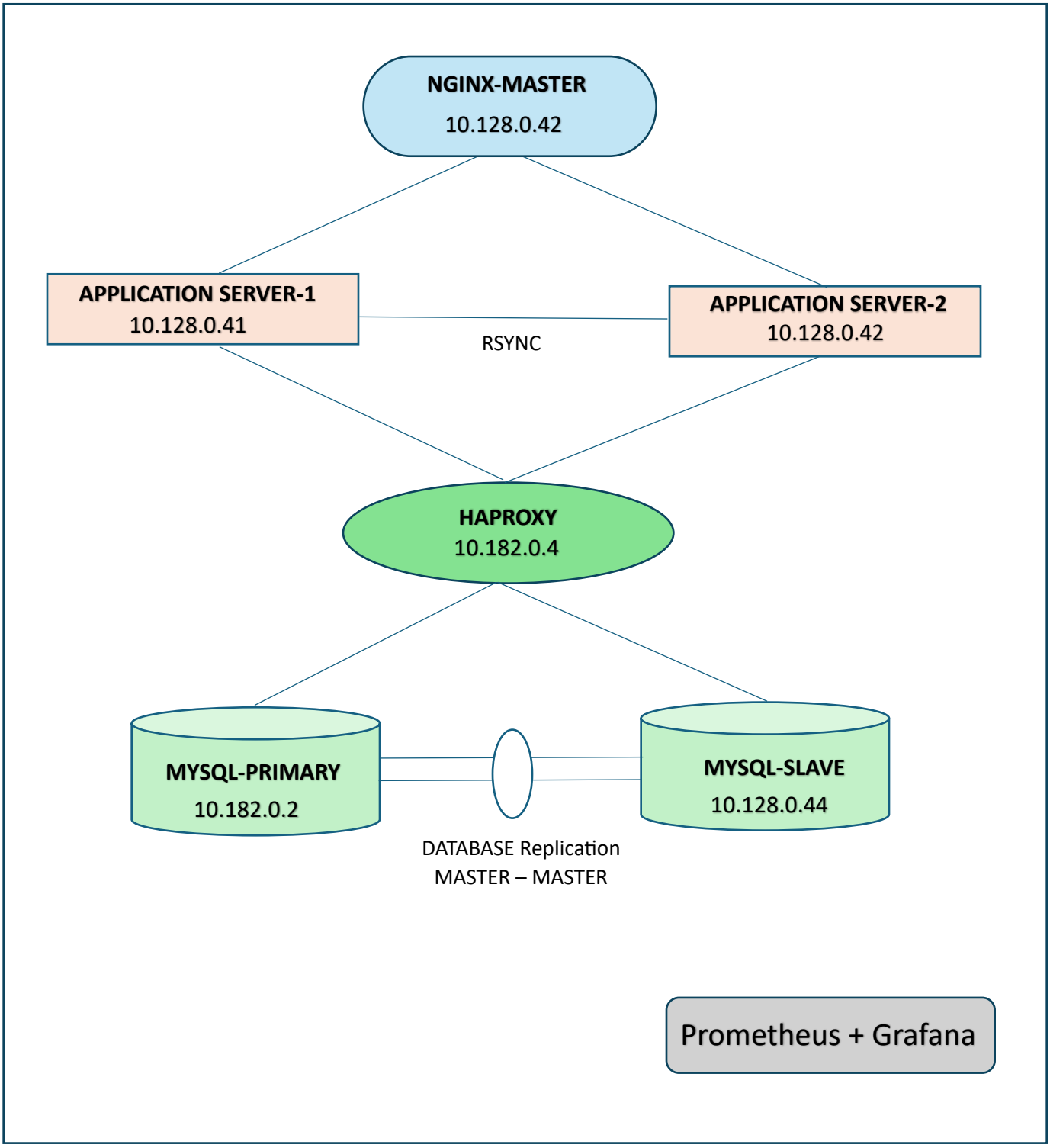
Scalable and Highly Available WordPress Deployment on Google Cloud



Overview of the Project

This project aims to implement a highly available, scalable, and fault-tolerant web application infrastructure on Google Cloud Platform (GCP). The architecture incorporates Nginx as a load balancer, reverse proxy multiple application servers for distributed processing, HAProxy for database load balancing, and a MySQL master-master replication setup to ensure data redundancy and high availability. To enhance observability and proactive monitoring, the infrastructure incorporates Prometheus for real-time metrics collection and Grafana for data visualization and alerting**.**

VM instances

Filter Enter property name or value

Status	Name	Zone	Recommendations	In use by	Internal IP
✓	appserver1	us-central1-c			10.128.0.41 (nic0)
✓	appserver2	us-central1-c			10.128.0.42 (nic0)
✓	haproxy	us-west4-b			10.182.0.4 (nic0)
✓	mysqlpromary	us-central1-c			10.128.0.43 (nic0)
✓	mysqlslave	us-west4-b			10.182.0.2 (nic0)
✓	nginxmaster	us-central1-a			10.128.0.44 (nic0)
✓	prometheus	us-west4-b			10.182.0.5 (nic0)

- ✓ Deploy **WordPress** on **Application Server 1** and **Application Server 2**, both running **Nginx** as the web server.
- ✓ **Application Server 1** and **Application Server 2** connect to **MySQL-Primary** and **MySQL-Slave** via **HAProxy** for database failover and high availability. HAProxy directs write operations to MySQL-Primary, while MySQL-Slave is used only during primary failure. **Master-Master** replication ensures **bidirectional synchronization**, allowing both servers to accept writes and maintain data consistency.
- ✓ **NGINX** is configured as a **reverse proxy** and **load balancer**, efficiently distributing incoming traffic between two application servers using the **round-robin algorithm** to ensure balanced load distribution and optimized performance.
- ✓ **Rsync** is employed to synchronize application code, configuration files, and static assets between Application Server 1 and Application Server 2, ensuring consistency across both servers.
- ✓ Strengthen **WordPress security** by implementing **authentication and authorization** for wp-admin, enforcing **advanced SSL configurations**, applying **rate limiting**, **hardening security headers**, and **enabling log monitoring** to mitigate unauthorized access, brute-force attacks, and other vulnerabilities.
- ✓ Set up **Prometheus and Grafana** for real-time **monitoring of all servers.**

Implementation Guide

Install and Configure Nginx, PHP, and WordPress on Appserver (10.128.0.41) /Appserver2 (10.128.0.42)

- Install NGINX and PHP on both Applicationserver1 and Applicationserver2
- Download and Deploy WordPress both Applicationserver1 and Applicationserver2
- Configure Nginx to Serve WordPress both Applicationserver1 and Applicationserver2 (/etc/nginx/conf.d/wordpress.conf)

  Note: Please refer attached configuration files,
  applicationserver1_wordpress.conf
  applicationserver2_wordpress.conf

Set Up MySQL on the Primary Database Server MySQL primary (IP Address 10.128.0.43)

- Install MariaDB on to MySQL primary server.
- Create the WordPress database and user
- Configure WordPress to Use the Remote MySQL Server (connect DB Host to MySQL Primary) (/var/www/html/wp-config.php)

Implement File Synchronization with RSYNC

- RSYNC is used for keeping application code, configuration files, and static assets in sync between appserver1 and appserver2.

Implement NGINX MASTER Server (IP Address 10.128.0.44)

- Install Nginx as a load balancer (Round Robin) and reverse proxy.

  There are several other load balancing mechanisms available, such as Least Connections, IP Hash, Weighted Load Balancing, Active health Checks (NGINX PLUS).

Load Test Check

```
[root@nginxmaster conf.d]# ab -n 10000 -c 100 http://10.128.0.44/
This is ApacheBench, Version 2.3 <$Revision: 1913912 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 10.128.0.44 (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests


Server Software:        nginx/1.20.1
Server Hostname:        10.128.0.44
Server Port:            80

Document Path:          /
Document Length:        169 bytes

Concurrency Level:      100
Time taken for tests:   1.332 seconds
Complete requests:      10000
Failed requests:        0
Non-2xx responses:      10000
Total transferred:      3590000 bytes
HTML transferred:       1690000 bytes
Requests per second:    7505.49 [#/sec] (mean)
Time per request:       13.324 [ms] (mean)
Time per request:       0.133 [ms] (mean, across all concurrent requests)
```

```
[root@nginxmaster chamath_lbr]# for i in {1..20}; do curl -s -I https://10.128.0.44/ | grep X-Upstream-Server; done
[root@nginxmaster chamath_lbr]# for i in {1..20}; do curl -s -I -H "Cache-Control: no-cache" https://techcha.xyz/ | grep X-Upstream-Server; done
X-Upstream-Server: 10.128.0.41:80
X-Upstream-Server: 10.128.0.42:80
X-Upstream-Server: 10.128.0.41:80
X-Upstream-Server: 10.128.0.42:80
X-Upstream-Server: 10.128.0.41:80
X-Upstream-Server: 10.128.0.42:80
X-Upstream-Server: 10.128.0.41:80
X-Upstream-Server: 10.128.0.42:80
X-Upstream-Server: 10.128.0.41:80
X-Upstream-Server: 10.128.0.42:80
X-Upstream-Server: 10.128.0.41:80
X-Upstream-Server: 10.128.0.42:80
X-Upstream-Server: 10.128.0.41:80
X-Upstream-Server: 10.128.0.42:80
X-Upstream-Server: 10.128.0.41:80
```

- Update the load balancer server block for **proxy Cache. Client-Side (Browser) Caching.**

There are several caching methods available, Such as FastCGI Caching, Client-Side (Browser) Caching, Microcaching,

Note: I attached **ngnixmaster configuration** file end of the document.

Cache Status Check

```
[root@nginxmaster conf.d]# curl -I https://techcha.xyz
HTTP/1.1 200 OK
Server: nginx/1.20.1
Date: Sun, 02 Mar 2025 15:46:04 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/8.0.30
Link: <https://techcha.xyz/wp-json/>; rel="https://api.w.org/"
X-Proxy-Cache: HIT
```

Implement MySQL Slave Server (IP Address 10.182.0.2)

- Install MySQL slave server for MASTER-MASTER database replication
- Create a Replication User for Bidirectional Replication for both MySQL-Primary and MySQL-Slave.
- Verify Replication on both servers

MySQL Primary

```
MariaDB [(none)]> SHOW SLAVE STATUS\G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 10.182.0.2
                  Master_User: replication
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000004
          Read_Master_Log_Pos: 7840759
               Relay_Log_File: mariadb-relay-bin.000006
                Relay_Log_Pos: 23701
        Relay_Master_Log_File: mysql-bin.000004
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
```

MySQL Slave

```
MariaDB [(none)]> SHOW SLAVE STATUS\G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 10.128.0.43
                  Master_User: replication
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000006
          Read_Master_Log_Pos: 749955
               Relay_Log_File: mariadb-relay-bin.000006
                Relay_Log_Pos: 485894
        Relay_Master_Log_File: mysql-bin.000006
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
```

Implement HAPROXY Server (IP Address 10.182.0.4)

- Create a Health Check User on Both MySQL Servers
- Configure Haproxy (/etc/haproxy/haproxy.cfg)

  Note: Please refer attached configuration files,
  haproxy.conf

- Configure HAPROXY IP (10.182.0.4) address for application server1 and application server2.
  Note: Please refer attached configuration files
  applicationserver1_wp-config.php

MySQL Primary

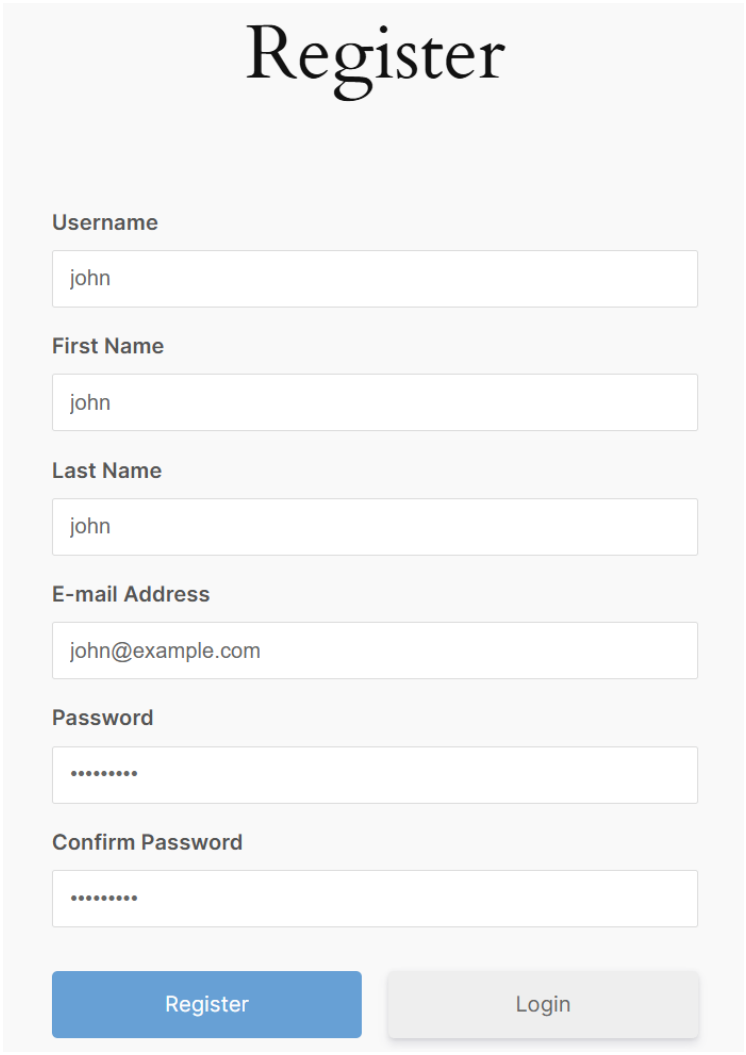
```
// ** Database settings - You can get this info from your web host **
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress_db' );

/** Database username */
define( 'DB_USER', 'wpuser' );

/** Database password */
define( 'DB_PASSWORD', 'wppassword' );

/** Database hostname */
define( 'DB_HOST', '10.182.0.4' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

Note: Please refer attached configuration files
applicationserver2_wp-config.php

MySQL Slave

```
// ** Database settings - You can get this info from your web host
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress_db' );

/** Database username */
define( 'DB_USER', 'wpuser' );

/** Database password */
define( 'DB_PASSWORD', 'wppassword' );

/** Database hostname */
define( 'DB_HOST', '10.182.0.4' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

- Verify **MASTER-MASTER** database replication & Haproxy



MySQL Primary

```
| 61 | john            | $P$B53sDdS77QIJW8tMJcTB3CMniiaRo3. | john           | john@example.com           |
phDqz85PfOzt0CaOmFyVjpGQAte1 |           0 | john john                    |
+----+-----------------+------------------------------------+----------------+----------------------------+
----------------------------+-------------+-----------------------------+
26 rows in set (0.000 sec)
```

MySQL Slave

```
| 61 | john            | $P$B53sDdS77QIJW8tMJcTB3CMniiaRo3. | john           | john@example.com
phDqz85PfOzt0CaOmFyVjpGQAte1 |           0 | john john                    |
+----+-----------------+------------------------------------+----------------+--------------------------
---------------------------+-------------+-----------------------------+
25 rows in set (0.000 sec)

MariaDB [wordpress_db]>
```

Test Case 01	Output
MySQL Primary and MySQL Slave both are running	Data writes to both MySQL primary and MySQL slave (via DB replication)

Test Case 02	Output
MySQL Primary down and MySQL Slave running	Data writes to MySQL Slave only

Test Case 03	Output
Once MySQL Primary UP	Replicate Data from MySQL slave to MySQL primary

Implement security mechanisms to enhance the overall security of the project

- Configure Authentication & Authorization for “wp-admin” on nginxmaster

```
    location ~* ^/wp-admin/.*\.php$ {

    auth_basic "Admin Login";
    auth_basic_user_file /etc/nginx/.htpasswd;
    proxy_pass http://wordpress_backend;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}

location /wp-login.php {

    auth_basic "Admin Login";
    auth_basic_user_file /etc/nginx/.htpasswd;
    proxy_pass http://wordpress_backend;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}
```

- Configure SSL Configurations.

```
server {
    listen 80;
    server_name techcha.xyz www.techcha.xyz;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl;
    server_name techcha.xyz www.techcha.xyz;

    # Access & Error Logs
    access_log /var/log/nginx/access.log custom_json_logs;
    error_log /var/log/nginx/error.log warn;

    ssl_certificate /etc/letsencrypt/live/techcha.xyz/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/techcha.xyz/privkey.pem;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers on;
    ssl_ciphers 'ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:...';
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;
```

- Rate Limiting

```
limit_req_zone $binary_remote_addr zone=login_zone:10m rate=10r/m;
limit_req_zone $binary_remote_addr zone=admin_zone:10m rate=20r/m;
limit_conn_zone $binary_remote_addr zone=conn_limit:10m;
```

```
    location ~* ^/wp-admin/.*\.php$ {

    auth_basic "Admin Login";
    auth_basic_user_file /etc/nginx/.htpasswd;
    limit_req zone=admin_zone burst=5 nodelay;
    proxy_pass http://wordpress_backend;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 }

 location /wp-login.php {

    auth_basic "Admin Login";
    auth_basic_user_file /etc/nginx/.htpasswd;
    limit_req zone=admin_zone burst=5 nodelay;
    proxy_pass http://wordpress_backend;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 }
```

- Security Header Hardening

```
# Security Headers for Hardening
add_header X-Frame-Options "SAMEORIGIN" always;
add_header X-Content-Type-Options "nosniff" always;
add_header X-XSS-Protection "1; mode=block" always;
add_header Referrer-Policy "strict-origin-when-cross-origin" always;
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload" always;
add_header Permissions-Policy "geolocation=(), microphone=(), camera=(), payment=()" always;
```

- Configure Gzip Compression for performance

```
gzip on;
gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;
gzip_vary on;
```

- Configure Custom Log Monitoring and Log rotate

```
# Access & Error Logs
access_log /var/log/nginx/access.log custom_json_logs;
error_log /var/log/nginx/error.log warn;
```

Access.log

```
[root@nginxmaster conf.d]# tail -f /var/log/nginx/access.log
13.38.119.25 - - [02/Mar/2025:17:02:33 +0000] "POST //www/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php HTTP/1.1" 301 169 "https://www.google.com/" "Mozilla
/5.0 (Macintosh; Intel Mac OS X 11_5_4) Gecko/20091401 Firefox/16.0" "-"
34.42.11.242 - - [02/Mar/2025:17:03:59 +0000] "POST /wp-cron.php?doing_wp_cron=1740935038.9561090469360351562500 HTTP/1.1" 301 169 "-" "WordPress/6.7.2; https://
techcha.xyz" "-"
34.42.11.242 - - [02/Mar/2025:17:05:23 +0000] "POST /wp-cron.php?doing_wp_cron=1740935122.896954059600830078125O HTTP/1.1" 301 169 "-" "WordPress/6.7.2; https://
techcha.xyz" "-"
34.42.11.242 - - [02/Mar/2025:17:06:40 +0000] "POST /wp-cron.php?doing_wp_cron=1740935200.133980989456176757812S HTTP/1.1" 301 169 "-" "WordPress/6.7.2; https://
techcha.xyz" "-"
34.42.11.242 - - [02/Mar/2025:17:07:45 +0000] "POST /wp-cron.php?doing_wp_cron=1740935264.832071065902709960937S HTTP/1.1" 301 169 "-" "WordPress/6.7.2; https://
techcha.xyz" "-"
34.42.11.242 - - [02/Mar/2025:17:08:50 +0000] "POST /wp-cron.php?doing_wp_cron=1740935329.705548048019409179687S HTTP/1.1" 301 169 "-" "WordPress/6.7.2; https://
techcha.xyz" "-"
{"time_local":"02/Mar/2025:17:09:17 +0000","remote_addr":"195.178.110.54","remote_user":"","request":"GET /cgi-bin/luci/;stok=/locale HTTP/1.1","status": "404","
body_bytes_sent":"113167","http_referer":"","http_user_agent":"","request_time":"3.545"}
204.76.203.15 - - [02/Mar/2025:17:09:23 +0000] "GET / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chr
ome/90.0.4430.85 Safari/537.36 Edg/90.0.818.46" "-"
118.139.177.200 - - [02/Mar/2025:17:10:14 +0000] "POST /xmlrpc.php HTTP/1.1" 301 169 "http://techcha.xyz" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7; rv:128
.0) Gecko/20100101 Firefox/128.0" "-"
{"time_local":"02/Mar/2025:17:10:14 +0000","remote_addr":"118.139.177.200","remote_user":"","request":"GET /xmlrpc.php HTTP/1.1","status": "403","body_bytes_sent
":"153","http_referer":"http://techcha.xyz","http_user_agent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7; rv:128.0) Gecko/20100101 Firefox/128.0","request_t
ime":"0.000"}
```

Logrotate

```
/var/log/nginx/*.log {
        daily
    rotate 14
    missingok
    compress
    delaycompress
    notifempty
    create 0640 nginx adm
    sharedscripts
    postrotate
        systemctl reload nginx > /dev/null 2>&1
    endscript
}

~
~
~
```

Note: Please refer attached configuration files,
nginxmaster_wordpress.conf

Implement Prometheus and Grafana for real-time monitoring of all servers

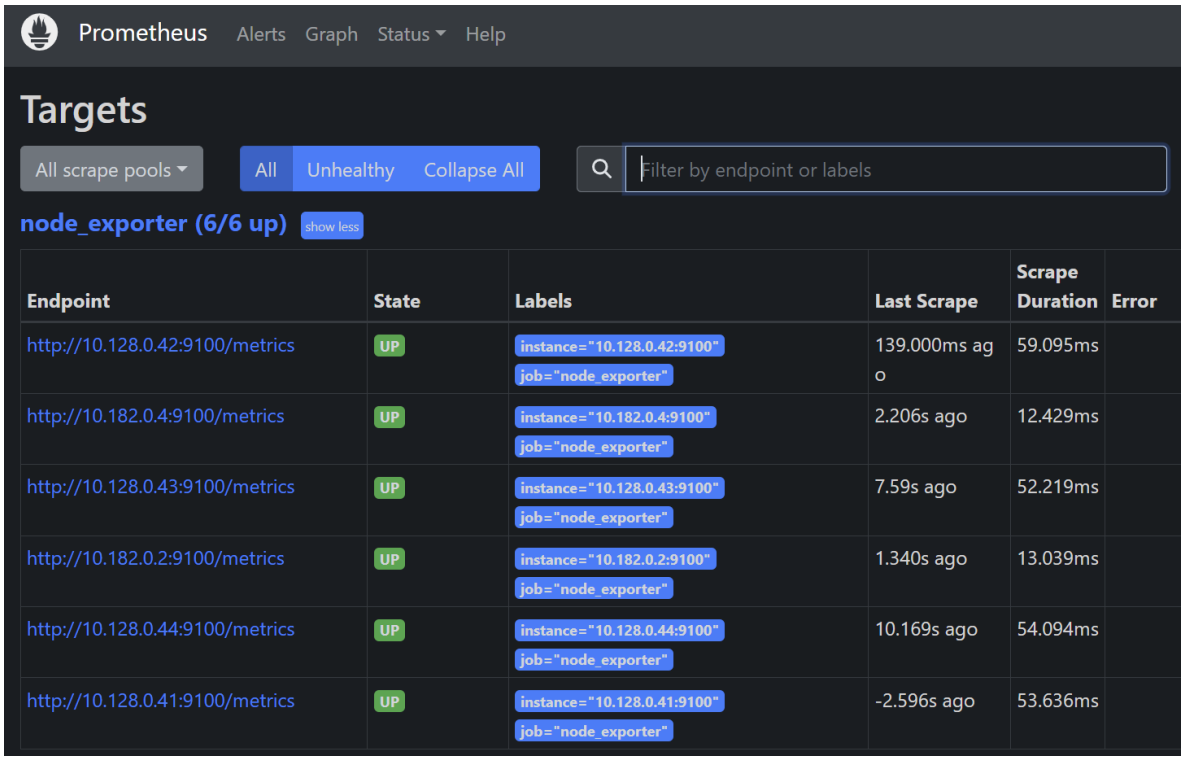
- Install & configure Prometheus on server (IP address 10.182.0.5)

  Note: Please refer attached configuration files,
  prometheus.yml
  prometheus. service

- Install and Configure Node Exporter on each server to collect CPU, memory, disk, and other OS-level metrics.

  Note: Please refer attached configuration files for appserver1,
  node_exporter.service

  Prometheus Dashboard



Grafana Dashboard