JOSEIT JOURNAL OF SYSTEMS ENGINEERING AND INFORMATION TECHNOLOGY
Published by: Indonesian Informatics Association
e-ISSN:2829-310X

Public IP Efficiency and Data Center Security Enhancement with Reverse Proxy Implementation

Achmad Sandy Bukhari[1*], Mohammad Iqbal[2]
[1]Master of Electrical Engineering, Gunadarma University, Jakarta, Indonesia
[2]Master of Technology and Engineering, Gunadarma University, Jakarta, Indonesia

*achmad.sandy@gmail.com

Abstract. With the increasing frequency of cyber-attacks, the trend of national cybersecurity traffic anomalies reached 976,429,996 incidents in 2022. Additionally, the world is now facing the fact that the supply of Public IPv4 addresses available for allocation is diminishing. IPv4 uses 32-bit addressing, which provides only over 4 billion unique IP addresses. By conducting research using two methods, namely a server without a reverse proxy and a server with an applied reverse proxy, it was found that implementing NGINX with a reverse proxy can lead to savings in public IPv4 addresses. Regardless of the number of servers, only one public IPv4 address is needed, which reduces the number of IPs required and also prevents cyber-attacks on the server. Testing with DNSChecker and whatismyipaddress showed that after applying the reverse proxy with NGINX, the application server could not be identified or accessed by external parties. Only the reverse proxy server was accessible to outsiders. As the number of applications increases, which directly correlates with the need for public IPv4 addresses, the study's results show that applying a reverse proxy with NGINX in a data center can overcome the limitations of public IPv4 addresses. As the number of virtual machines and applications grows, a single public IPv4 address applied to the reverse proxy server suffices. Thus, implementing a reverse proxy with NGINX allows multiple servers to use just one public IPv4 address.

Keywords: *IPv4, Public IP, Reverse Proxy, NGINX, Cyber Attack*

Cite as: Sandy Bukhari, A., & Iqbal, M. (2024). Public IP Efficiency and Data Center Security Enhancement with Reverse Proxy Implementation. Journal of Systems Engineering and Information Technology (JOSEIT), 3(2), 37-42. https://doi.org/10.29207/joseit.v3i2.5948
Received by the Editor: 24-07-2024
Final Revision: 25-09-2024
Published: 30-09-2024

1. Introduction

Public IP addresses are IPv4 addresses managed and allocated by IDNIC-APJII to Internet Service Providers (ISPs). According to the website https://idnic.net/, IDNIC-APJII manages both IPv4 and IPv6 addresses. It's important to note that currently, IPv4 addresses are limited worldwide. All Data Centers in Indonesia use public IPv4 addresses provided by IDNIC-APJII. Due to the limited availability of IPv4 addresses, it is considered necessary to conserve public IPv4 addresses. The background of the problem is as follows, in Figure 1.1, based on information from the CERT (Computer Emergency Response Team) published on the website https://idnic.net/service/cert, it was found that in February 2022, Distributed Denial of Service (DDoS) attacks/Flooding were the highest trend, with a total of 1,487 incidents reported. DDoS attacks are a type of flooding attack aimed at rendering computer or network resources unavailable. The method involves sending a large volume of excessive requests or traffic to a server.

With the increasing number of applications that need to be facilitated in the Data Center, in line with the Presidential Regulation regarding SPBE (Electronic Goods and Services Procurement), the demand for public IP addresses also rises. The government is required to build Data Center infrastructure that meets SPBE standards, encompassing security, efficiency, and sustainability. Without effective solutions, such as the use of Reverse Proxy with NGINX, Data Centers will struggle to provide sufficient public IP addresses, potentially hindering the implementation and operation of new applications. This implementation will ensure that new applications can run smoothly and securely.

Furthermore, the use of a single public IP address for multiple servers through the implementation of Reverse Proxy with NGINX can lead to significant savings. This solution allows one public IP address to be used for various servers, thus reducing the need for numerous public IP addresses. These savings are crucial not only from a cost perspective but also in terms of resource management, enabling Data Centers to operate more efficiently and effectively . Given the scarcity of IPv4 addresses, this solution supports operational efficiency and ensures compliance with applicable regulations, guaranteeing that Indonesia's technological infrastructure can grow safely and sustainably.

Zhongcheng Lei [11] discusses the utility of NGINX Reverse Proxy in providing uninterrupted services without the necessity of adding domain names or HTTPS certificates. This study utilizes the Open-Source NGINX, as detailed by NGINX (2023), which serves as an HTTP and reverse proxy server, email proxy server, and a versatile TCP/UDP proxy server developed by Igor Sysoev [22]. Previous studies have highlighted the effectiveness of NGINX in optimizing server performance and security [9]. Against the backdrop of global constraints on IPv4 Public IP addresses and the escalating frequency of cyber-attacks on governmental bodies, including DDoS assaults, Darknet Exposure, and other forms of cyber intrusions, the imperative arises to optimize the use of Public IP addresses and fortify Data Center security through the deployment of an NGINX-based Reverse Proxy Server. Implementation of such a server is anticipated to bolster server performance, ensure seamless application operations, and elevate the quality of public services. The rationale for this research stems from several key objectives; the deployment of a Reverse Proxy Server utilizing NGINX, the potential for one Public IPv4 address to serve multiple servers, the enhancement of server security, the cost-effectiveness of server operations, and the optimal utilization of Public IPv4 addresses. This study aims to enhance Public IP Efficiency, augment Server Security and Efficiency, and optimize infrastructure costs associated with Public IPs, thereby indirectly streamlining governmental expenditures on server management infrastructure.

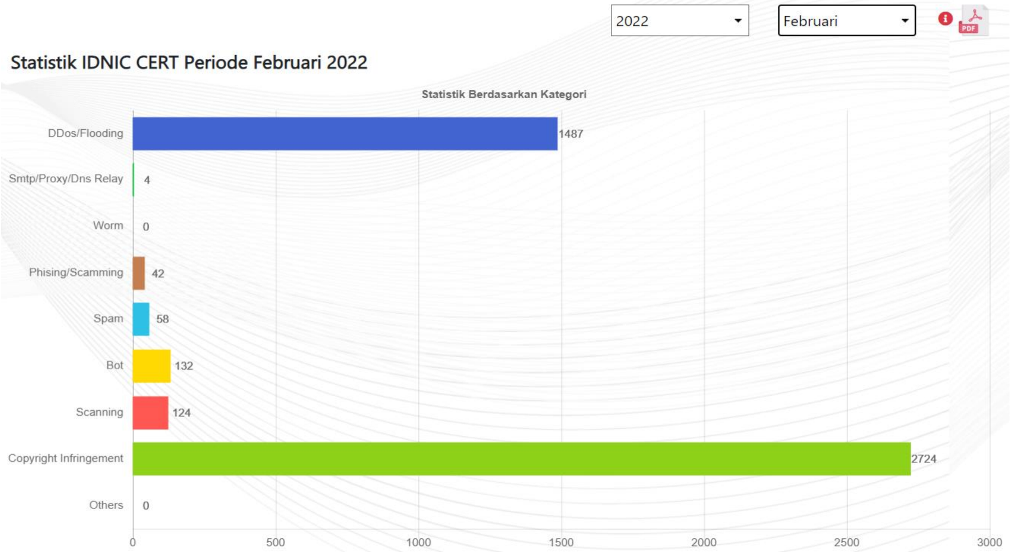


Figure 1. IDNIC CERT Statistics for February 2022

The initial phase of the research involved a comprehensive literature review. This stage focused on gathering and analyzing existing literature pertinent to the research topic. The sources included both national and international journals. The literature review was essential to identify best practices and to uncover theories relevant to VMware, Reverse Proxy, Network Topology, and other related areas necessary for the study. This extensive review helped in understanding the current state of knowledge and the methodologies previously employed in similar research. Based on the findings from the literature review, the infrastructure requirements were outlined. This involved determining the necessary servers and their specifications, the network topology to be utilized, and the overall architecture of the infrastructure to be developed.

The process included identifying the optimal hardware and software components that would support the research objectives, ensuring that the proposed solutions would be both efficient and effective. This step was crucial in setting the groundwork for the practical implementation phase of the study. The designed infrastructure architecture was then translated into a detailed business process flow. This architectural blueprint served as a visual and conceptual guide, illustrating how different components of the system would interact and function together. It outlined the sequence of operations, data flow, and the integration of various technological elements. This clear depiction of the infrastructure was essential for ensuring that all aspects of the research were aligned with the study's goals, facilitating a smoother transition from theoretical planning to practical execution.

In this research, two websites were utilized for testing purposes: http://bansos1.kotabogor.go.id/ and http://bansos2.kotabogor.go.id/. These websites served as test cases to examine the impact of the Reverse Proxy implementation on Public IP visibility and accessibility. The differences in the test results highlighted the efficacy of the Reverse Proxy in obscuring the server details from external parties, thus enhancing security [19]. The use of these specific websites provided concrete examples to support the findings, demonstrating the practical benefits of the Reverse Proxy implementation in a controlled environment. This empirical evidence was crucial in validating the theoretical insights gained from the literature review and establishing the practical applicability of the research outcomes [7]. As the number of applications increases, the demand for Public IP AS Numbers also rises correspondingly. The findings of this research indicate that implementing a Reverse Proxy with NGINX in a Data Center can effectively address the limitations posed by the scarcity of Public IP AS Numbers. By leveraging the capabilities of a Reverse Proxy Server using NGINX, a single Public IP AS Number can be utilized across multiple servers, thereby optimizing the allocation of IP resources. This solution is particularly beneficial as the number of Virtual Machines and applications grows, ensuring that only one Public IP AS Number is necessary for the entire server infrastructure [4].

After implementing the Reverse Proxy in the server environment using NGINX, significant differences were observed in the test results of the Public IP using the Ping Command tool and DNS Checker [12]. This research utilized two specific websites for testing: http://bansos1.kotabogor.go.id/ and http://bansos2.kotabogor.go.id/. The test results demonstrated the effectiveness of NGINX Reverse Proxy in masking the true IP addresses of the servers, thereby enhancing security and reducing the likelihood of cyber-attack [10]. The empirical evidence gathered from these tests underscores the practical advantages of using a Reverse Proxy in a server environment. One of the most notable benefits of using NGINX is the ability to achieve substantial Public IPv4 address savings. With the implementation of NGINX, a single Public IP address can be used to serve multiple servers, which is a significant advantage in the context of limited IP resources. This approach not only improves the efficiency of IP address utilization but also contributes to cost savings and better resource management. By adopting NGINX Reverse Proxy, organizations can optimize their server infrastructure, enhance security, and make more efficient use of available Public IP addresses.

2. Methods

The method used in this research involves identifying the server infrastructure needs, followed by the installation and configuration of the server, and subsequently configuring the DNS server. The initial phase of the research is the Literature Review, which involves studying literature related to the research title. This stage of the research involved practical application and empirical testing to evaluate the effectiveness of the Reverse Proxy. The literature review provided a theoretical foundation, while the practical implementation allowed for real-world validation of the concepts. This combination of theoretical and practical approaches ensured a comprehensive understanding of the subject matter.

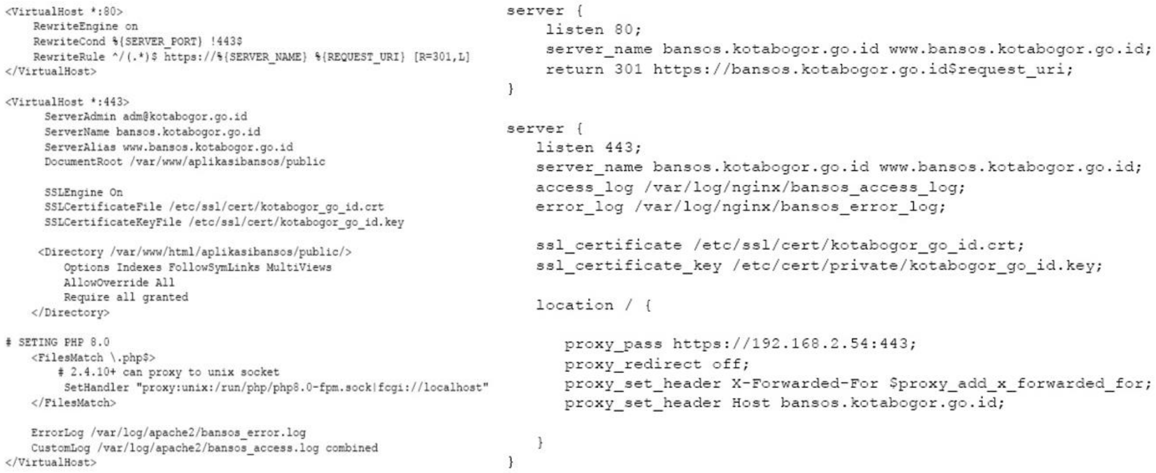
Based on the findings from the literature review, the infrastructure requirements were outlined, including the necessary servers and their specifications, the network topology to be used, and the architecture of the infrastructure to be developed. The infrastructure architecture was represented as a business process flow. The next step involved defining the specifications for Virtual Machines (VM) on VMware. This phase included creating specifications for the VM servers on VMware: Application Server, Reverse Proxy Server, and DNS Server. These specifications encompassed CPU capacity, RAM, HDD, as well as the local IP for the application server, the public IP for the Reverse Proxy Server, and the DNS Server. In addition to these hardware specifications, the required software for each of the three servers was also determined. The servers were to run on Linux operating systems, with the application server hosting a database and web server applications, and Nginx being used as a reverse proxy on the Reverse Proxy Server.

The subsequent step was the installation and configuration of the Application Server. This phase included the following processes: creating the VM server with the planned specifications (CPU cores, RAM, HDD), installing the operating system, configuring the local IP, and installing the necessary web server applications such as Apache, PHP, and databases like MySQL or others. Following this, the application files and database were uploaded to the server, the database was created, and user access rights were assigned. The database was then imported, and the database configuration settings were applied to the application. Once the application settings were completed, the next step was configuring the virtual host to ensure the application had a domain that could be accessed even with just a local IP. This included configuring the virtual host settings.

The installation and configuration of the Reverse Proxy Server followed, which involved installing the operating system, installing Nginx, and configuring the virtual host. Since the domain name would use the IP of this Proxy Server, in addition to setting the local IP, a public IP was also added. In the /etc/nginx/conf.d directory, virtual host settings for the application domain were configured. One Reverse Proxy Server can handle multiple connections to application servers. For security, a firewall such as firewall-cmd and fail2ban was installed. Next was the configuration of the DNS Server. The DNS Server was already existing and in use, so this step primarily involved configuring the subdomains on the DNS Server. Subdomain settings on the DNS Server included the public IP of the Reverse Proxy Server and the subdomain names. To verify if the subdomains were correctly pointing to the Reverse Proxy IP, checks could be performed using https://dnschecker.com. Successful subdomain resolution by global DNS servers could be confirmed if the public IP of the Reverse Proxy Server appeared when the subdomain name was entered.

2.1 Creating a Subdomain Management Model Using Reverse Proxy.

The creation of a subdomain management model using a reverse proxy server is carried out through the following stages: Setting up virtual hosts on the application server, configuring virtual hosts on the reverse proxy server, and configuring subdomains on the DNS server.

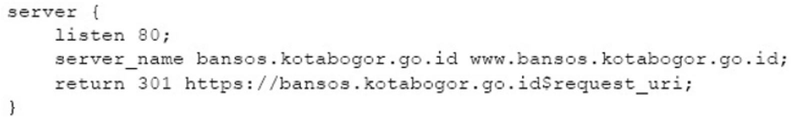


```
<VirtualHost *:80>
    RewriteEngine on
    RewriteCond %{SERVER_PORT} !443$
    RewriteRule ^/(.*)$ https://%{SERVER_NAME} %{REQUEST_URI} [R=301,L]
</VirtualHost>

<VirtualHost *:443>
      ServerAdmin adm@kotabogor.go.id
      ServerName bansos.kotabogor.go.id
      ServerAlias www.bansos.kotabogor.go.id
      DocumentRoot /var/www/aplikasibansos/public

      SSLEngine On
      SSLCertificateFile /etc/ssl/cert/kotabogor_go_id.crt
      SSLCertificateKeyFile /etc/ssl/cert/kotabogor_go_id.key

     <Directory /var/www/html/aplikasibansos/public/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        Require all granted
    </Directory>

# SETING PHP 8.0
    <FilesMatch \.php$>
       # 2.4.10+ can proxy to unix socket
        SetHandler "proxy:unix:/run/php/php8.0-fpm.sock|fcgi://localhost"
    </FilesMatch>

    ErrorLog /var/log/apache2/bansos_error.log
    CustomLog /var/log/apache2/bansos_access.log combined
</VirtualHost>
```

```
server {
    listen 80;
    server_name bansos.kotabogor.go.id www.bansos.kotabogor.go.id;
    return 301 https://bansos.kotabogor.go.id$request_uri;
}

server {
   listen 443;
   server_name bansos.kotabogor.go.id www.bansos.kotabogor.go.id;
   access_log /var/log/nginx/bansos_access_log;
   error_log /var/log/nginx/bansos_error_log;

   ssl_certificate /etc/ssl/cert/kotabogor_go_id.crt;
   ssl_certificate_key /etc/cert/private/kotabogor_go_id.key;

   location / {

      proxy_pass https://192.168.2.54:443;
      proxy_redirect off;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host bansos.kotabogor.go.id;

   }
}
```

Figure 2. The following program lines on Virtual host settings on the application server and virtual host settings on the reverse proxy server

There are several coding blocks, each of which regulates a specific function:

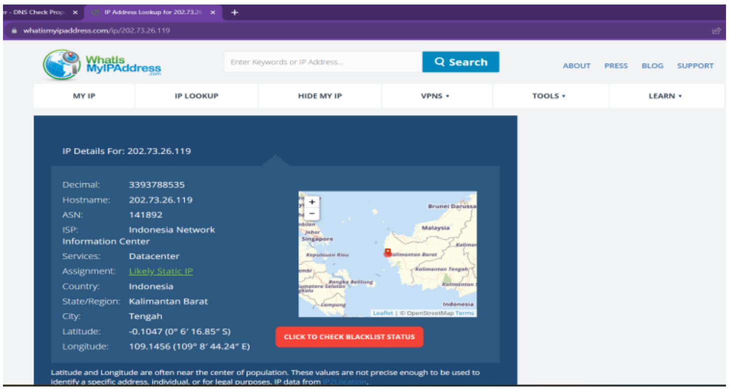


```
server {
    listen 80;
    server_name bansos.kotabogor.go.id www.bansos.kotabogor.go.id;
    return 301 https://bansos.kotabogor.go.id$request_uri;
}
```

The coding block in Figure declares a proxy server that functions as a reverse proxy, forwarding client access to the domain bansos.kotabogor.go.id. This access is directed to the proxy server, which then forwards it to the application server with the local IP address 192.168.2.54. The settings on the DNS (Domain Name Server) function to forward client requests accessing the subdomain bansos.kotabogor.go.id. The DNS server will translate this subdomain access request to its public IP address. In this case, the public IP provided by the DNS server is the public IP of the reverse proxy server: 202.102.15.3.

3. Results and Discussion

After implementing Reverse Proxy on the Server Environment using NGINX, there are differences in the results of checking the Public IP using the Ping Command tool and DNSChecker. In this research, two websites are used: http://bansos1.kotabogor.go.id/ and http://bansos2.kotabogor.go.id/. Figure 3.3 shows the results of checking the subdomain http://bansos1.kotabogor.go.id/ using https://dnschecker.org/, indicating that the subdomain can be accessed from San Francisco America, Mountain View America, Berkeley America, Burnaby Canada, Melbourne Australia, Singapore, Seoul South Korea, Shenzhen China, Indian cities, Islamabad Pakistan, Ireland, and Dhaka Bangladesh.



IP Details For: 202.73.26.119

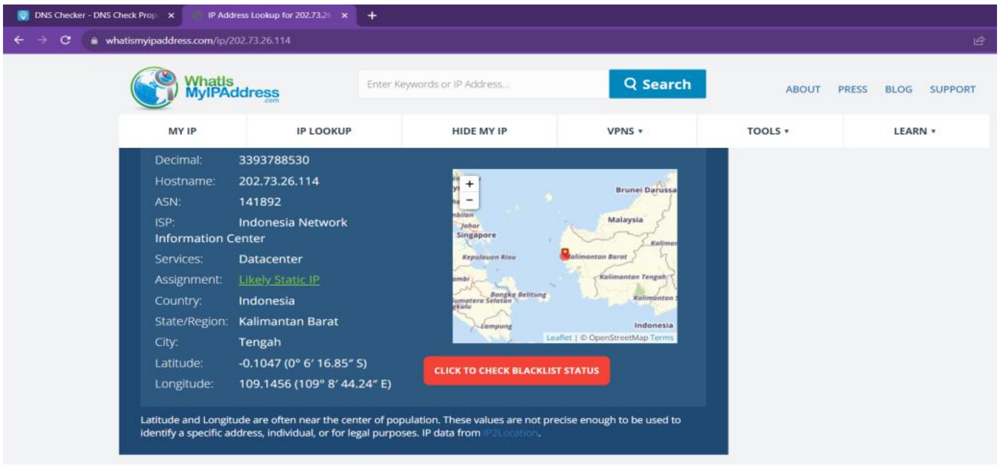
Decimal: 3393788535
Hostname: 202.73.26.119
ASN: 141892
ISP: Indonesia Network Information Center
Services: Datacenter
Assignment: Likely Static IP
Country: Indonesia
State/Region: Kalimantan Barat
City: Tengah
Latitude: -0.1047 (0° 6′ 16.85″ S)
Longitude: 109.1456 (109° 8′ 44.24″ E)

Latitude and Longitude are often near the center of population. These values are not precise enough to be used to identify a specific address, individual, or for legal purposes. IP data from IP2Location.

Figure 3. Detail examination of the public IP address 202.73.26.119

The public IP for http://bansos1.kotabogor.go.id/ is also obtained as 202.73.26.119. Following this, using the tool https://whatismyipaddress.com/ provides detailed data results of this public IP in Figure 3. To find out the IP route, a tool in the form of tracert is used. Tracert is a command to show the route a packet takes to reach its destination. The mechanism is carried out by sending an Internet Control Message Protocol (ICMP) Echo Request message to the destination with an increasing time-to-live value.

Figure 4 shows the results of the examination of the subdomain http://bansos2.kotabogor.go.id/ using https://dnschecker.org/, indicating that the subdomain can be accessed from San Francisco America, Mountain View America, Berkeley America, Miami America, Melbourne Australia, Singapore, Seoul South Korea, Shenzhen China, City India, Islamabad Pakistan, Ireland, and Dhaka Bangladesh. Additionally, the public IP for bansos1.kotabogor.go.id is found to be 202.73.26.114. Using this public IP, further details of the public IP data are obtained using the tool https://whatismyipaddress.com/ as shown in Figure 4.



Decimal: 3393788530
Hostname: 202.73.26.114
ASN: 141892
ISP: Indonesia Network Information Center
Services: Datacenter
Assignment: Likely Static IP
Country: Indonesia
State/Region: Kalimantan Barat
City: Tengah
Latitude: -0.1047 (0° 6′ 16.85″ S)
Longitude: 109.1456 (109° 8′ 44.24″ E)

Latitude and Longitude are often near the center of population. These values are not precise enough to be used to identify a specific address, individual, or for legal purposes. IP data from IP2Location.

Figure 4. Detail examination of the public IP address 202.73.26.114

On the subdomain https://bansos1.kotabogor.go.id/ , settings without Reverse Proxy have been implemented, resulting in the public IP of the server being 202.73.26.119. On the second VM server, on the subdomain https://bansos2.kotabogor.go.id/ , Reverse Proxy using NGINX has been applied, and the public IP of the server is found to be 202.73.26.114. This demonstrates that NGINX is effectively working to conceal the IP address of the application server https://bansos2.kotabogor.go.id/.

4. Conclusions

Through testing using DNSChecker, it was found that on the server employing the method Without Reverse Proxy, the subdomain https://bansos1.kotabogor.go.id/ is publicly accessible. Using the tool https://whatismyipaddress.com/ reveals that the public IP is 202.73.26.119. This public IP, 202.73.26.119, is the IP address of the application server bansos1. Therefore, without the implementation of Reverse Proxy, external parties can directly access the physical application server. Unlike the bansos1 application server, the bansos2 server has implemented the Reverse Proxy method. Through testing using DNSChecker, it was found that the subdomain https://bansos2.kotabogor.go.id/ is publicly accessible. Using the tool https://whatismyipaddress.com/ reveals that the public IP is 202.73.26.114.

This public IP, 202.73.26.114, is the IP address of the Reverse Proxy server. After implementing Reverse Proxy with NGINX, it was found that the bansos2 application server cannot be identified or accessed by external parties. Instead, external parties can only access the Reverse Proxy server. By adding applications that are directly proportional to the need for public IP AS Numbers, the research findings show that implementing Reverse Proxy with NGINX in the Data Center can overcome the constraint of limited public IP AS Numbers. With the addition of Virtual Machines and Applications, it is sufficient to use only 1 (one) public IP AS Number applied to the Reverse Proxy Server. Therefore, by implementing Reverse Proxy with NGINX, multiple servers can utilize just 1 (one) public IP.

References

[1] Christinger Tomer (2017). Cloud Computing and virtual machines in LIS education: options and resources. Emerald Insight, Vol.33, No. 1, 2017

[2] M. Agung Nugroho dan Cuk Subiyantoro (2018). Analisis Cluster Container pada Kubernetes dengan Infrastruktur Google Cloud Platform. JIPI, Vol 3, No. 2, 2018.

[3] Nugroho MA. (2018) Analisis Cluster Container Pada Kubernetes Dengan Infrastruktur Google Cloud Platform. JIPI (Jurnal Ilm Penelit dan Pembelajaran Inform. 2018;

[4] Carisimo E, Selmo C, Alvarez-Hamelin JI, Dhamdhere A. (2019). Studying the evolution of content providers in IPv4 and IPv6 internet cores. Comput Commun.;145:54–65.

[5] Esteban Cairisimo (2019). Studying the evolution of content providers in IPv4 and IPv6 inernet cores. ScienceDirect. Computer Communication 145 (2019) 54-65.

[6] Abd Al Ghaffar (2020). Government Cloud Computing and National Security, Emerald insight. Januari 2020.

[7] DeJonghe D. Nginx cookbook. O'Reilly Media; 2020

[8] Rizki Agung Muzaki (2020). Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall. IEEEE. Vol. 978-1-7281-9098-3/20.

[9] Jusuf Qarkaxhija (2020). Using Cloud Computing as an Infrastructure Case Study- Kwun-Hung Li dan Kon-Yeung Wong(2021). Empirical Analysis of IPv4 and IPv6 Networks through Dual-Stack Sites. MDPI. Vol. 12. Microsoft Azure. TECHNIUM, Vol. 2, Issue 3 ff.93-100, 2020

[10] Muzaki RA, Briliyant OC, Hasditama MA, Ritchi H. Improving security of web-based application using ModSecurity and reverse proxy in web application firewall. In: 2020 International Workshop on Big Data and Information Security (IWBIS). IEEE; 2020. p. 85–90

[11] Zhongcheng Lei (2020). Cost-Effective Server-side Re-deployment for Web-based Online Laboratories Using NGINX Reverse Proxy. ScienceDirect. IFAC PapersOnLine 53-2 (2020) 17204–17209.

[12] Qarkaxhija J. Using cloud computing as an infrastructure case study-microsoft azure. 2020

[13] Borislac Dordevic (2021). VMware ESXi and Microsoft Hyper-V Hypervisor Performance Comparison. IEEE Xplore

[14] Chen Ma dan Yuhong Chi (2022). Evaluation Test and Improvement of Load Balancing Algorithms of Nginx. IEEEAccess. Digital Object Identifier 10.1109/ACCESS.2022.3146422

[15] Derek Dejonghe (2022). Nginx Cookbook, Advanced Recipes for High Performance Load Balancing. O'REILLY. Available on : https://www.nginx.com/resources/library/complete-nginx-cookbook/#download

[16] Goparaju B, Rao BS. A DDoS Attack Detection using PCA Dimensionality Reduction and Support Vector Machine. Int J Commun Networks Inf Secur. 2022;14(1s):1–8.

[17] Muhanad Rafli (2022). Pengujian Kinerja Load Balancing Web Server Menggunakan Nginx Riverse Proxy berbasis OS Centos 7. Jurnal Teknik Informatika dan Sistem Informasi. Vol. 9, No. 3, September 2022, Hal. 1824-1840.

[18] Muhammad Garzali Qabasiyu (2022). Use of VMware Virtualization technology to Deploy Private Cloud Computing Infrastructure as A Service On Business Organizations. IJRCS. Vol. 06, Issue-01, Januari 2022.

[19] Rafli M. Jurnal Pengujian Kinerja Load Balancing Web Server menggunakan Nginx Riverse Proxy Berbasis OS Centos 7. JATISI (Jurnal Tek Inform dan Sist Informasi). 2022;9(3):1824–40

[20] Andrii Chyrvon (2023). The Main Methods of Load Balancing On The NGINX Web Server. Scientific Practice: Modern and Classical Research Methods

[21] Chyrvon A, Lisovskyi K, Kyryndas N. the Main Methods of Load Balancing on the Nginx Web Server. Collect Sci Pap «ΛΌΓΟΣ». 2023;(May 26, 2023; Boston, USA):146–51.

[22] Igor Sysoev, NGINX (2023). Penjelasan NGINX [Online]. Available at: https://nginx.org/en/ [Accessed 1 Oktober 2023]