ADVANCED SOFTWARE ARCHITECTURES SOEN 6471

SUMMER 2023

Deliverable 1

NGINX

Declaration

We, the members of the team, have read and understood the Fairness Protocol and the Communal Work Protocol, and agree to abide by the policies therein, without any exception, under any circumstances, whatsoever.

Team L

Abhimanyu Sharma - 40188795

Kirthana Senguttuvan - 40189746

Saurabh Sharma - 40226298

Urvish Rupeshbhai Tanti - 40197508

Vaibhav Sharma - 40197697

Table of Contents

List of Figures

1. Description of NGINX

1.1. Introduction

NGINX [1] is a high-performance open-source web server that was first released in 2004. It was created by Igor Sysoev, a Russian software engineer, and has since become one of the most popular web servers on the internet. NGINX is pronounced "engine x" and is known for its speed, scalability, and low memory usage.

1.2. Purpose

The purpose of nginx is to serve as a web server that can handle high traffic and requests. It is designed to be lightweight and efficient, making it an ideal choice for modern website architecture. Nginx can also be used as a reverse proxy server [2], load balancer [3], and HTTP cache [4].

1.3. Characteristics

One of the defining characteristics of nginx is its ability to handle large numbers of concurrent connections with minimal resources. This makes it an excellent choice for websites that experience high levels of traffic or need to handle many requests at once. Nginx achieves this through its event-driven architecture, which allows it to handle multiple connections simultaneously without creating new threads or processes.

Another unique feature of Nginx is its modular design. Nginx consists of a core module that handles basic web server functionality such as serving static files and processing HTTP requests. Additional modules can be added to provide additional functionality such as load balancing, caching, or SSL/TLS encryption [5].

1.4. Development

Nginx has been developed using the C programming language and has been optimized for performance on modern hardware architectures. The development team behind Nginx has focused on creating a web server that is fast, reliable, and easy to configure.

In terms of software activities carried out in the development of nginx, the project has been open-source from the beginning. This means that anyone can contribute code or suggest improvements to the project. The community around Nginx has grown over time and includes developers from all over the world who have contributed bug reports, suggestions, and observations that have helped improve the software. The priority of the development activities is on performance optimizations, refining algorithms, reducing latency, and enhancing scalability to ensure Nginx performs efficiently under heavy workloads.

1.5. Summary

Nginx’s remarkable performance, scalability, and efficiency have made it a reliable web server and reverse proxy software. Its event-driven architecture [6], caching capabilities, and load-balancing functionalities make it an ideal choice for handling high-traffic web applications. The ongoing software activities of Nginx's development team, including core development, feature enhancements, bug fixing, and performance optimization, combined with the contributions of its open-source community, ensure Nginx remains a reliable and flexible solution for web server needs.

2. Model for Context of Use

NGINX is a versatile and widely adopted web server and reverse proxy server that caters to the needs of various user groups.

- Website Owners
- Developers
- System Administrators
- DevOps Professionals
- CDN Providers
- Hosting Companies
- Networking Professional

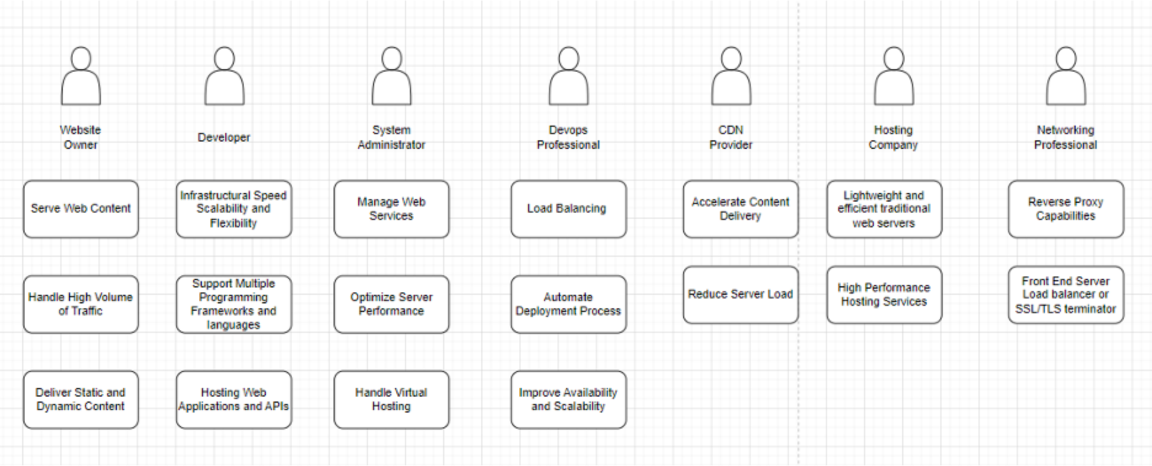

Website Owner
Developer
System Administrator
Devops Professional
CDN Provider
Hosting Company
Networking Professional
Serve Web Content
Infrastructural Speed Scalability and Flexibility
Manage Web Services
Load Balancing
Accelerate Content Delivery
Lightweight and efficient traditional web servers
Reverse Proxy Capabilities
Handle High Volume of Traffic
Support Multiple Programming Frameworks and languages
Optimize Server Performance
Automate Deployment Process
Reduce Server Load
High Performance Hosting Services
Front End Server Load balancer or SSL/TLS terminator
Deliver Static and Dynamic Content
Hosting Web Applications and APIs
Handle Virtual Hosting
Improve Availability and Scalability

Figure 1: Use Cases of different users from NGINX

2.1. Website Owners

As a website proprietor, it is essential to possess a dependable platform to serve your web content, and NGINX provides that precisely. It is an optimal solution for catering to your diverse needs while seamlessly delivering pages, images, CSS files, and JavaScript files to visitors. The platform is adept at handling high volumes of traffic by acting as a reverse proxy server [10] and load balancer [11], distributing incoming requests across multiple backend servers and ensuring optimal resource utilization to prevent any single server from becoming overwhelmed.

NGINX is capable of efficiently delivering both static and dynamic content, making it the ideal choice for website owners who are keen on providing a smooth user experience for visitors. With NGINX, you can effectively manage and scale your website, regardless of the content's complexity or the traffic load's height. In summary, NGINX is an invaluable tool that enables website proprietors to deliver their content reliably, efficiently, and with optimal resource utilization.

2.2. Developers

Developers find NGINX to be a valuable tool for their projects due to its speed, scalability, and flexibility. With NGINX, developers can achieve high-performance web applications by efficiently handling incoming requests and delivering content quickly. NGINX's architecture allows for easy scalability, enabling developers to handle increasing traffic loads without compromising performance. Additionally, NGINX supports multiple programming languages and frameworks, making it versatile and adaptable to developers' preferences and requirements. This flexibility allows developers to leverage NGINX regardless of the technology stack they are using. Furthermore, NGINX is well-suited for hosting web applications and APIs, providing reliable and secure delivery of these services to end-users. Its robust features and comprehensive documentation make it a popular choice among developers for building and deploying their applications.

2.3. System Administrators

System administrators play a crucial role in managing web servers, and NGINX is a powerful tool that they rely on to streamline their tasks. They utilize NGINX to efficiently manage web servers [5] by configuring and fine-tuning server settings, ensuring smooth operation, and optimal performance. NGINX empowers system administrators to optimize server performance by leveraging features like load balancing, caching, and connection pooling, which enhance resource utilization and handle high traffic loads effectively. Additionally, NGINX enables system administrators to handle various server-related tasks, including virtual hosting, SSL/TLS termination, access control, and more. With NGINX, system administrators can effortlessly set up and manage virtual hosts, secure web communications with SSL/TLS termination, and enforce access control measures, making it an indispensable tool for their server management responsibilities [8].

2.4. DevOps Professionals

NGINX is widely embraced by DevOps professionals for its ability to streamline and enhance various aspects of their work. One key benefit is its load balancing capabilities, which enable DevOps professionals to distribute incoming network traffic efficiently across multiple servers, ensuring optimal resource utilization and preventing server overload. Moreover, NGINX empowers DevOps professionals to automate development processes by seamlessly integrating with tools like CI/CD pipelines and containerization platforms. This automation significantly accelerates the deployment and delivery of applications, enabling faster iteration and reducing time-to-market. Additionally, NGINX plays a crucial role in improving availability [6] and scalability for DevOps professionals. By acting as a reverse proxy server and effectively managing backend server resources, NGINX enhances the overall availability of applications. Its scalability features allow DevOps professionals to dynamically scale their infrastructure to meet changing demands, providing a reliable and responsive user experience. Overall, NGINX empowers DevOps professionals to optimize their workflows, enhance application availability, and efficiently scale their systems [9].

2.5. CDN Providers

CDN Providers, as users of NGINX, greatly benefit from its capabilities to accelerate content delivery and reduce server load. By leveraging NGINX's caching functionality, CDN Providers can store frequently accessed content closer to end-users, strategically placing servers in multiple locations worldwide. This proximity allows for faster content retrieval and reduced latency, improving the overall user experience. Additionally, NGINX's ability to efficiently handle high traffic loads and distribute requests across multiple servers enables CDN Providers to effectively manage and balance server loads, preventing overload situations and maintaining optimal performance. Overall, NGINX empowers CDN Providers to enhance content delivery speed, scalability, and reliability while efficiently utilizing server resources.

2.6. Hosting Companies

NGINX is an excellent choice for hosting companies seeking a lightweight and efficient alternative to traditional web servers. With its streamlined architecture and optimized performance, NGINX enables hosting companies to efficiently serve web content to their clients. By utilizing NGINX as their primary web server, hosting companies can ensure high-performance hosting services that can handle heavy traffic loads with ease. NGINX's ability to handle concurrent connections and efficiently manage resources allows hosting companies to deliver fast and reliable hosting services to their customers, resulting in an enhanced user experience and improved customer satisfaction.

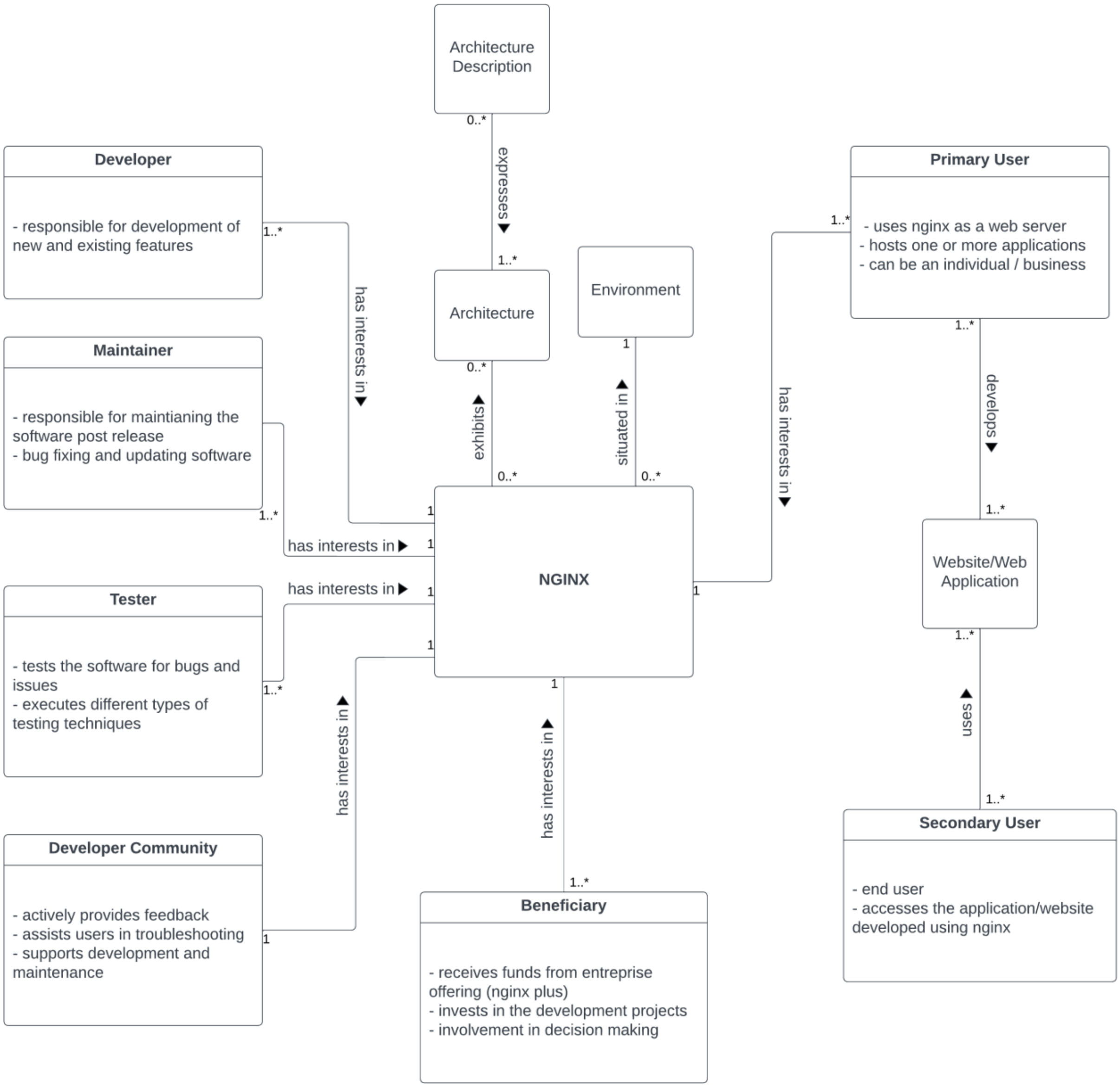
2.7. Networking Professionals

NGINX is highly valued by networking professionals for its versatile capabilities as a reverse proxy. As a front-end server, NGINX efficiently handles incoming client requests, allowing networking professionals to distribute traffic across multiple backend servers using its robust load balancing capabilities. With NGINX's ability to act as a load balancer, networking professionals can achieve optimal resource utilization, scalability, and high availability for their applications. Additionally, NGINX serves as an SSL/TLS terminator, enabling networking professionals to offload the encryption and decryption processes from backend servers. This not only enhances security but also improves overall performance by reducing the computational burden on the backend infrastructure. These features make NGINX an indispensable tool for networking professionals seeking to enhance the reliability, scalability, and security of their network infrastructure.

3. Stakeholder Model

3.1. Class Diagram

After analyzing the architecture and identifying the stakeholders from [1] and [14], the following stakeholder model [15] in the form of UML class diagram can be constructed for NGINX. The diagram follows the **ISO/IEC/IEEE 42010** standard [16].



Architecture Description
0..*
expresses ▼
1..*
Architecture
0..*
exhibits ▲
0..*
Environment
1
situated in ▲
0..*
Developer
- responsible for development of new and existing features
1..*
has interests in ▼
1
Maintainer
- responsible for maintianing the software post release
- bug fixing and updating software
1..*
has interests in ▶
1
Tester
- tests the software for bugs and issues
- executes different types of testing techniques
1..*
has interests in ▶
1
Developer Community
- actively provides feedback
- assists users in troubleshooting
- supports development and maintenance
1
has interests in ▲
1
NGINX
1
has interests in ▲
1..*
Beneficiary
- receives funds from entreprise offering (nginx plus)
- invests in the development projects
- involvement in decision making
1
has interests in ▼
1..*
Primary User
- uses nginx as a web server
- hosts one or more applications
- can be an individual / business
1..*
develops ▼
1..*
Website/Web Application
1..*
uses ▲
1..*
Secondary User
- end user
- accesses the application/website developed using nginx

Figure 2: Class diagram representing the stakeholders in the NGINX software [17]

3.2. Stakeholder Descriptions

- **Developer**: This stakeholder participates in the development and improvement of NGINX's functionality, the addition of new features, and codebase maintenance.

- **Maintainer**: The primary responsibilities of this stakeholder are to respond to bug reports, resolve software problems, and make sure that NGINX is stable and reliable after it has been launched.

- **Tester**: This stakeholder oversees applying various testing methodologies to NGINX in order to find and report any defects or issues, hence ensuring the software's quality and accuracy.

- **Developer Community**: A group of developers who take an active interest in the NGINX ecosystem make up this stakeholder. They offer suggestions, contribute code, help users troubleshoot issues, and support NGINX's continuing development and upkeep.

- **Beneficiary**: This stakeholder is an organization or business that gains from NGINX. They provide funds to NGINX development initiatives, choose investments, and influence the company's strategic course.

- **Primary User**: This participant uses NGINX as a web server to host one or more applications and is the main user of it. The main user is anyone who uses NGINX to serve web content effectively and securely, whether they are an individual or a company.

- **Secondary User**: This stakeholder group includes people or organizations who use NGINX-powered programs or webpages. As end users, they communicate with the NGINX-based systems and use the principal user's services.

Team Contributions

Group/Team/Unit Member	Contributions
Abhimanyu Sharma	• Identified Stakeholders and their roles by reviewing NGINX documentations (for Problem 3). • Searched on ISO/IEC/IEEE 42010 standard for architecture description. • Created Figure 1 – UML Class Diagram for Stakeholder Mode by analyzing how stakeholder classes are linked and their respective cardinalities.
Kirthana Senguttuvan	• Performed a detailed analysis on the unique characteristics of NGINX such as load balancing, reverse proxy capabilities, the effectiveness of content caching, the importance of SSL/TLS encryption, and the need for event-driven architecture. • Researched the purpose of NGINX, development activities performed by the team and open-source contributors, task prioritization, and the evolution of software over time. • Consolidated the overall information gathered briefly to address Problem-1.
Saurabh Sharma	• Researched and defined the use case context of NGINX, including its role as a web server, reverse proxy, load balancer, and SSL/TLS terminator. • Defined the context of use for various user groups, such as website owners, developers, system administrators, DevOps professionals, CDN providers, hosting companies, and networking professionals, with regards to NGINX.
Urvish Rupeshbhai Tanti	• Identified Stakeholders and their role by reviewing documentations for Problem 3 • Provided definitions for the stakeholders
Vaibhav Sharma	• Produced context for use models on NGINX by reviewing use case documentation on NGINX through open source and third party of articles

References

[1] “The Architecture of Open Source Applications (Volume 2)nginx,” aosabook.org. https://aosabook.org/en/v2/nginx.html.
[2] “What is a Reverse Proxy Server? | NGINX,” NGINX, 2018. https://www.nginx.com/resources/glossary/reverse-proxy-server/
[3] NGINX, “What Is Load Balancing? How Load Balancers Work,” NGINX, 2018. https://www.nginx.com/resources/glossary/load-balancing/
[4] “NGINX Content Caching | NGINX Documentation,” docs.nginx.com. https://docs.nginx.com/nginx/admin-guide/content-cache/content-caching/.
[5] “What is SSL/TLS Encryption?” https://www.f5.com/glossary/ssl-tls-encryption#:~:text=TLDR%3A%20SSL%2FTLS%20encrypts%20communications
[6] E. Pangu, “Nginx Event Driven Architecture — Demonstrated in Code,” Medium, Jan. 15, 2021. https://levelup.gitconnected.com/nginx-event-driven-architecture-demonstrated-in-code-51bf0061cad9.
[7] “How nginx processes a request” nginx.org/en/docs/http/request_processing.html
[8] “What are the use cases for NGINX Plus?” https://www.nginx.com/faq/what-are-the-use-cases-for-nginx-plus/
[9] “F5 NGINX Management Suite” https://www.nginx.com/products/nginx-management-suite/
[10] “NGINX Reverse Proxy | NGINX Documentation” https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/
[11] “TCP and UDP Load Balancing | NGINX Documentation” https://docs.nginx.com/nginx/admin-guide/load-balancer/tcp-udp-load-balancer/
[12] “Configuring NGINX and NGINX Plus as a Web Server” https://docs.nginx.com/nginx/admin-guide/web-server/web-server/
[13] “High Availability Support for NGINX Plus in On-Premises Deployments” https://docs.nginx.com/nginx/admin-guide/high-availability/ha-keepalived/
[14] NGINX Product Documentation https://docs.nginx.com/
[15] Lecture notes - ‘STAKEHOLDERS OF SOFTWARE ARCHITECTURE’ by Prof. Pankaj Kamthan
[16] ISO/IEC/IEEE 42010:2022 Software, systems and enterprise — Architecture description https://www.iso.org/standard/74393.html
[17] Class diagram created using Lucidchart - https://www.lucidchart.com/