https://doi.org/10.46335/IJIES.2024.9.7.1 e-ISSN: 2456-3463
Vol. 9, No. 7, 2024, PP. 1-4

International Journal of Innovations in Engineering and Science, www.ijies.net

GAMEOPS

Reema Roychaudhary[1], Nishant Gharpure[2], Vidhi Shiyani[3], Shrinidhee Pande[4], Rushi Wagh5

[1]Assistant Professor, [2,3,4]Student Department of Computer Engineering, SVPCET, Nagpur
,
rroychaudhary@stvincentngp.edu.in

***Received on**: 5 May,2024* ***Revised on**: 29 June,2024* ***Published on**: 02 July ,2024*

***Abstract—** GameOps, derived from the union of "Game" and "DevOps," symbolizes the merging of Game development and DevOps practices. The primary objective of this research is to create a collaborative online gaming experience that merges DevOps principles with game development strategies, aiming to reduce the time required to launch a website-based cooperative game. This involves integrating DevOps and Game Development concepts, such as Continuous Integration and Delivery, to achieve a swift Time to Market (TTM). The paper documents the approach to construct a captivating multiplayer gaming portal that ensures a cooperative gaming environment. The study examines the interaction between these interconnected domains.*

Keywords— Unity, Continuous integration, Continuous deployment, Cooperative games, DevOps

I. INTRODUCTION

In today's technology sector, companies need to deliver their products to clients as quickly as possible to remain competitive. This means that companies need to focus on achieving minimal time to market for their products or services, this can be achieved through various software development methodologies such as continuous integration and continuous deployment. Before DevOps, the software development life cycle (SDLC) used to take a prolonged release of software due to manual testing, bug detection, and manual deployment [1]. These conventionally occurring obstacles can be disentangled using Ansible, Docker, and Jenkins automation [2]. With DevOps practices, companies can rationalise their development process and deliver software to customers faster and with minimal errors [3]. These DevOps practices have been used to integrate Game development and Web development domains.

The website is a platform to host the game, it is an interface between the user and the game. The user can register and log in to the account and access the game. The website is integrated with Firebase which is a database tool. Firebase is a database service provided by Google. database is used to store user information and user login credentials. Firebase is better than any other conventional database service because it does not require code or SQL queries and is easy to set up.

The game in question is an online cooperative multiplayer(co-op) game. A co-op game allows players to work together against various challenges and opponents [4]. Players control a spaceship and try to destroy enemies while completing a specified task which is determined by the different game modes that can be selected by the players.

II. LITERATURE REVIEW

A. *Primary Goal*

The primary objective was to create an easily accessible and fun experience. This was achieved by developing a website-based game. A multiplayer component was added to enhance the gaming experience by allowing multiple players to participate simultaneously [5]. To support the game, a suitable website hosting platform was also required. Therefore, the project demanded a comprehensive development environment and a method to seamlessly integrate these environments.

B. *Identification*

Today's gaming landscape is dominated by multiplayer experiences, seamlessly connecting players and fostering a sense of community. These games can be played through various network connections, including the Internet and other networks. Multiplayer games have become an industry norm [6]. Hence the multiplayer feature has been incorporated in the game.

The most common method to access the internet is through the browser. Hence the website makes the game more accessible and saves the players from the hassle of download and installation of the game.

The game as well as the website require vastly different development environments. These environments need to be integrated to ensure proper development of both the features. The integration is accomplished using the principles of DevOps [7]. These principles can be easily implemented using the Amazon Web Services (AWS) platform.

III. METHODOLOGY

The game was developed using the Unity engine, a robust tool for creating and publishing games on various platforms. Unity also offers various services tailored for implementing multiplayer games.

Players interact with games in different ways, primarily through Player versus Environment (PvE) and Player versus Player (PvP). In a PvE game, players face a computer-controlled environment, while in a PvP game, players compete against each other. Additionally, there are Player versus Player versus Environment (PvPvE) games, where players encounter both computer-controlled threats and real players. This project aims to create a PvE experience where players must collaborate to achieve a common goal.

Unity provides various features to streamline the game development process such as a Physics system, UI modules and multiplayer packages. The game development process started with the planning of the essentials systems and their interactions with each other. Planning was followed by the implementation of a single-player prototype for the game. The singleplayer prototype was then expanded upon to implement the multiplayer. The various systems implemented for the game include:

a. Input System: Input system is an input manager provided by unity. It is used to streamline the input management between different input devices such as Keyboard/Mouse and controllers.

b. Lobbies: A lobby is a collective space or a staging area for players to gather and/or wait for game to start. The lobby feature has been implemented using the Unity Gaming Services (Unity Cloud).

c. Relay: Relay is another service provided by Unity Cloud. Relay service is used to connect players to the server seamlessly. It saves the player from thee hassle of manually opening ports to connect to a server.

React.js is typically used to create websites. A JavaScript package called React.js is used to create web apps and user interfaces. However, React.js lacks integrated routing and features client-side rendering. Moreover, authentication is not supported by default. Next.js is a react framework which has been used to build the website. It enables the developer to create a scalable website with next.js since it is significantly faster. It ensures improved performance through server-side rendering and static site generation. Additionally, it offers simplified routing and data fetching with integrated TypeScript support. Tailwind CSS is used for website styling because of its utility-first approach, which enables developers to quickly style components by applying pre-defined utility classes straight to the JSX component. Next.js easily connects with Tailwind CSS which makes it easier to configure.

Next.js, the newest and most effective technology available at present, was used to build the website. With the help of this React framework, developers are able to create fast and scalable websites. In order to ensure faster page loads and better performance, Next.js enables developers to render React components on the server side before sending them to the client. This server-side rendering capability is built-in, making it easy to implement without additional configuration. Each React component in the pages directory instantly becomes a route in the application as Next.js also employs a file-based routing system. This method streamlines routing and facilitates managing and organizing pages with ease. Simple, effective, and adaptable, Next.js is well-known for these qualities.

DevOps seamlessly integrates the various aspects of the game and the website. DevOps offers a fully automated approach to software development and deployment through its integrated toolchain (ITC). The ITC is a chain of tools that streamlines the development and deployment of software, making the process efficient and effective. Tools used in ITC are as follows:

a. Unity Engine: The Unity engine provides a simple was used to design and develop various game systems.

b. Unity Cloud: Unity Cloud provides essential game services needed to implement a multiplayer game. It has been used to use services for lobby and player connection

c. GitHub: GitHub is a powerful Source Code Management (SCM) tool used to keep track of the various changes that go on during the development process.

d. Firebase: Firebase is database service provided by Google. Firebase has been used to store the user data. Data is stored in a JSON like format which is a NoSQL database program.

e. Amazon Web Services: By leveraging Amazon Elastic Compute Cloud (Amazon EC2), we were able to effortlessly adjust our computing resources based on the varying demands of our operations. This allowed us to maximize our efficiency and productivity while minimizing any unnecessary costs.

f. Docker: Docker gives developers an effective and efficient way to deploy customised images of a container, docker image file consists of the installation of Apache HTTPD webserver and pulling source code from Jenkins workspace to the webserver directory (/var/www/html)

g. Ansible: Ansible playbook is used for the configuration of the load balancer and proxy server

h. Jenkins: Jenkins is an integration tool, used to extract code from GitHub and copy it into the workspace.

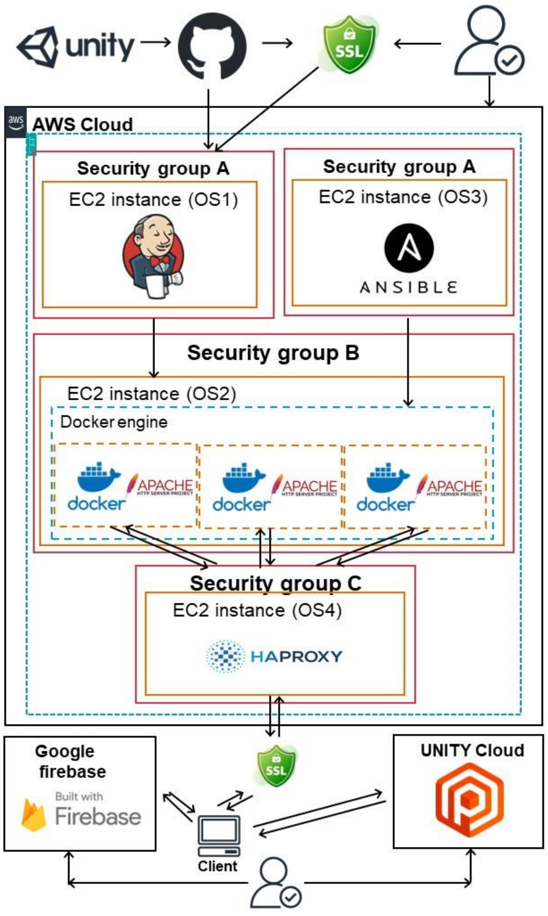


Fig 1. Pipeline

The steps involved in ITC are as follows:

1. Create a Website Graphics Library (WebGL) build for the game using the Unity game engine.
2. Integrate the website and WebGL build.
3. Test build website on the local host.
4. Push code to GitHub repository.
5. Install Jenkins in AWS instance (OS1).
6. Pull code in Jenkins workspace. Make a directory in the slave node. Copy the workspace directory in the slave node (AWS instance OS2). Run docker commands to install the webserver package (HTTPD) and restart the web server.
7. Install Ansible in AWS instance (OS3).
8. Run ansible-playbook for configuration of HAproxy Package.
9. Run ansible-playbook for configuration of the load balancer

To ensure secure communication between instances, OS1 and OS3 belong to security group A, which only permits inbound requests from OS2. Similarly, OS2 is part of security group B, which allows it to communicate with OS1, OS3, and OS4. Additionally, the website has been assigned SSL certification and a domain name to ensure the security of user data.

IV. DATABASE

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real-time. Every linked client receives real-time data synchronization in JSON format from the data storage. Backend-as-a-Service (BaaS) app development platform Firebase offers hosted backend services including cloud storage, hosted static files, real-time database, authentication, and authentication.

The website and the game require three types of user data. This data is used to authenticate users, track their progress in the game, and store essential game data.

The website uses a Firebase database to store important user information, such as usernames, email addresses, and passwords. The game also accesses, uses, and modifies this data, as well as stores its game-specific data in the database. The specific data required by the website includes:

1. Username: The name that the user will use in the game.
2. Email: The email address that the user uses to log in to the website.

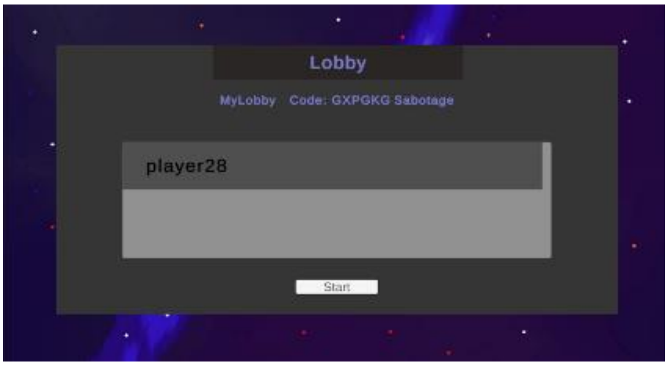


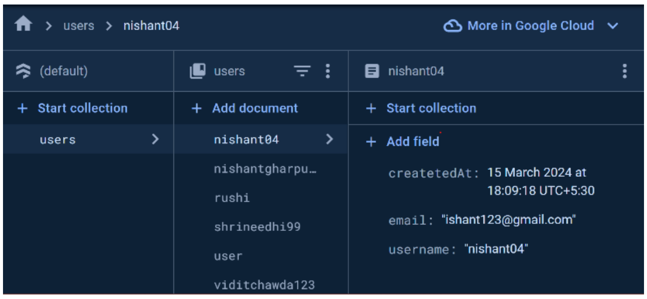
Fig 2. Firebase Database

V-KEY CHALLENGES

1. Configuration of the load balancer.
2. Integration of game WebGL with login page
3. Lobby creation for users.
4. Connecting users with different IP addresses
5. Proper synchronization of game data to ensure lag-free gameplay.

VI. RESULTS

1. Players need to create or join a lobby to play the game. A lobby which has been created can be seen in the Figure 3.
2. Build triggers (Poll SCM) are used to fetch any update from the GitHub repository and is deployed in one minute.
3. Load balancer has been successfully implemented using Ansible-Playbook.

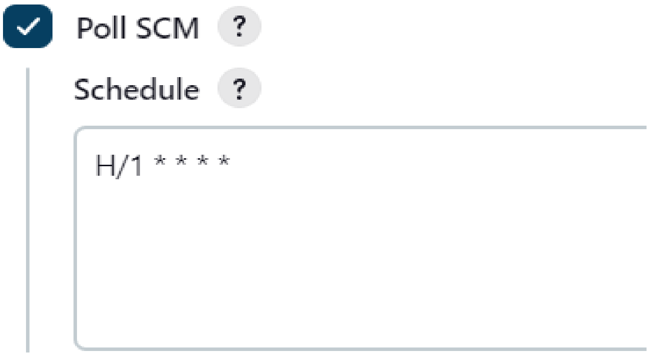


users

nishant04
nishantgharpu…
rushi
shrineedhi99
user
viditchawda123

createtedAt: 15 March 2024 at 18:09:18 UTC+5:30
email: "ishant123@gmail.com"
username: "nishant04"

Fig 3. Multiplayer lobby


☑ Poll SCM ?

Schedule ?

H/1 * * * *

Fig 4. Jenkins Poll SCM

VII. CONCLUSION

This paper showcases the remarkable synergy between DevOps principles and game development methodologies, culminating in the creation of a collaborative website-based game. Our goal of minimising time to market is upheld by the practices of Continuous Integration (CI) and Continuous Delivery (CD). Through this collaborative approach, we achieved the development of a robust gaming platform and enhanced the user experience in online multiplayer gaming. The insights gained from this project serve as a testament to the effectiveness of merging these disciplines, paving the way for future innovations at the intersection of technology and gaming.

This project has been designed and developed to be extensible. New features can be added and delivered to the users efficiently. Additionally, the data from website can be analysed to further enhance the user experience. These additional features might include improvements to the existing content as well as addition of new content.

REFERENCES

[1] *Poonam and Pooja Mittal, DevOps Tools at different stages of Software Development: Analysis and Review, National Conference on Emerging Trends in Smart Computing, ETSC-2019, Rohtak, ISBN 978-93-80544-35-9*

[2] *Poonam Narang, Pooja Mittal, Hybrid Model for Software Development: An Integral Comparison of DevOps Automation Tools, Indonesian Journal of Electrical Engineering and Computer Science (IJEECE), IAES Publishers, ISSN 2502-4752, Scopus, SJR 2020 (Q3 0.241), Vol 27, No 1, July 2022, pp 456-465.*

[3] *Ramdin Jabbari, Nauman Bin Ali, Binish Tanveer, Kai Petersen (2016), what is DevOps? A Systematic Mapping Study on Definitions and Practices, ACM Digital Library, published in Proc. of The Scientific Workshop Conference XP2016.*

[4] *Quandt, Thorsten; Kröger, Sonja (2014). Multiplayer: The Social Aspects of Digital Gaming. London: Routledge. ISBN 978-0-415-82886-4.*

[5] *Mulligan, Jessica; Patrovsky, Bridgette (2003). Developing online games: an insider's guide. Indianapolis, Ind. [u.a].: New Riders Publ. ISBN 978-1-59273-000-1.*

[6] *Bagga, Atul, October 2011. Emerging Trends In Games-as-a-Service. Game Developers Conference.*

[7]*Bolscher R, Daneva M. (2019), Designing software architecture to support continuous delivery and DevOps: a systematic literature review, ICSOFT. 2019: 27-39.*