

Module 8-1 Project Three Final

Valerie J. Smith

Southern New Hampshire University

CS-230-H4756 Operating Platforms 21EW4

Dr. Vivian Lyon

April 24, 2021



Draw It Or Lose It
CS 230 Project Software Design Document
Version 1.3

Table of Contents

CS 230 Project Software Design Document	2
Table of Contents	3
Document Revision History	3
Executive Summary	4
Design Constraints	4
System Architecture View	5
Domain Model	5
Recommendation Strategy Comparisons	13
Recommendations	20

Document Revision History

Version	Date	Author	Comments
1.0	03/20/2021	Valerie J. Smith	The Gaming Room initial design document and high-level recommendations
1.1	03/28/2021	Valerie J. Smith	Provide high level overview of details for the Recommendations
1.2	04/11/2021	Valerie J. Smith	Provide architectural recommendations and incorporate feedback
1.3	04/24/2021	Valerie J. Smith	Incorporate additional final feedback for Linux support benefits

Executive Summary

The Gaming Room has requested a web-based solution to extend their current gaming application, Draw It Or Lose It. The game is currently offered as an Android application. The web-based version of the game will feature the current functionality of the current Android application.

A high-level overview of a solution would be to create a web application version of the game, utilizing a Linux cloud server for the back end, and a client facing web application in the form of a Single Web Page application, created with ReactJS or Angular JS. It may also be of benefit to the client to have the many images needed stored in a CDN, or Content Delivery System as well.

For the Linux server, a recommendation is to utilize a platform such as Amazon Web Services, where the client would be able to set their requirements as well as receive additional training documentation for their employees. Amazon Web Services offers computes created with games as the focus (Amazon Services, 2021).

Due to the security concerns of the client, it is recommended to use the authentication and authorization capabilities that are also found in Amazon Web Services (Amazon Web Services I. , 2021).

Design Constraints

Technical Constraints:

- Current operating system, software, and hosting platform of the client is unknown and will need to be determined in order to support the new web application.
- The location of the images and other assets that are required for development has yet to be determined.
- The skill set of the client's employees is inexperienced for developing web applications
- Training will need to be obtained by the client in order to support the new web application.
- Authentication and authorization for the game user login and capabilities will need to be developed in order to secure the application.

Business Constraints

- The timeline for project completion has not been determined.
- The budget for this project has not been proposed.
- Copyright requirements for the images used and any logos/artwork are needed.

System Architecture View

Please note: There is nothing required here for these projects, but this section serves as a reminder that describing the system and subsystem architecture present in the application, including physical components or tiers, may be required for other projects. A logical topology of the communication and storage aspects is also necessary to understand the overall architecture and should be provided.

Domain Model

The Gaming Room class diagram is a structural class diagram that supports Object Oriented programming principals such as portability, inheritance, encapsulation, and polymorphism. The Gaming Room program domain model contains seven classes, with the main class residing in the Driver class to start the program, and an additional Test class for testing purposes. The main classes support portability as they are each self-contained objects, for example, a Game.

The Entity class is a class that demonstrates inheritance as it is being inherited by the Game, Team, and Player class. The abstraction that the Entity class provides is the ability for each class that extends Entity to create an id and a name, as well as extend the use of the ability to use the getId and getName functions to retrieve that stored data. The Entity class acts as a manager for the id and name variables, which are private to this class. This removes the responsibility of creating the id and name variables over again for each extending class.

The Game class features an overloaded constructor that has the parameters of id and Name. The overloaded constructor is an example of Object-Oriented programming principle of polymorphism, where the Game class extends the Entity class. The Game class also contains an Array List named Team. The Array List Team is named teams, and is private to the Game class. The Game class contains an addTeam function that takes a parameter of 'name', which allows the class to add a team to the Array List of teams

by name, if the name does not already exist. There can be zero or more teams created from the Game class.

The Team class extends the Entity class, and contains an Array List of type Player, which is named players. The Team class also contains an overloaded constructor and an addPlayer method that allows a player to be added to the players Array List by Name if the name does not already exist. The Team class can create zero or more players.

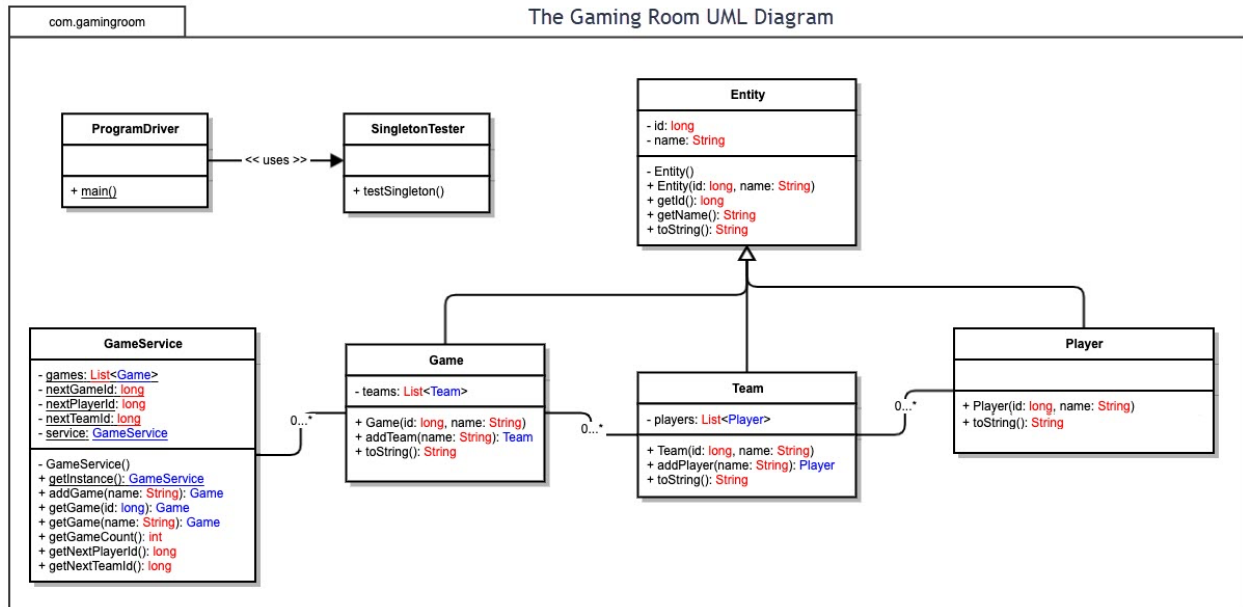
The Player class extends the entity class and contains an overloaded constructor named Player, which has the parameters of Id and name. The Player class will create one player by id and name.

The classes of Entity, Game, Team, and Player also contain the toString method, which is used to represent an object as a String and is often used to print object contents. Overriding the toString method returns details about the object and saves additional coding.

The GameService class has the ability to create an instance of zero or more games. The GameService class is a singleton class, which characteristics contain a private instance of the service, a private default constructor, and a public static method getInstance that returns one and only one instance of the game service. This is important for the client in order to ensure that there is only one instance of any one game in play at a time.

The GameService contains get functions in order to get game, next player Id, and next team Id. Encapsulation of the variables being set to private and static ensures that a game cannot be created outside of the class.

The object-oriented approaches of Portability, Encapsulation, and Polymorphism are created in a way that will ensure that the code is more secure, which fulfills the software design requirements.



Evaluation			
Development Requirements			
Server Side			
Mac	Linux	Windows	Mobile Devices
<p>Server side utilizes Xgrid to adhoc Mac systems, designed for large clusters. (Apple, n.d.) Apple offers the Apache web server, which can be configured for a group of Mac computers, or hosted on a platform such as Amazon Web Services.</p> <p>Advantages: An advantage of Mac OS Server is to simplify the distribution of complicated tasks for Mac Systems. The Mac OS Server can be run from a Mac computer and is good for small businesses. The Apache Web Server is open source and free to use with their license. It is cross platform and works on Unix and Windows servers, as well as web browsers such as Chrome, Firefox, Safari (Domantas, 2021).</p> <p>-Unix OS -Open Source servers -Secure - User and file access controls</p>	<p>Server-side features, such as hosting ,utilize open-source software, giving multiple hosting options.</p> <p>Advantages: Much safer than other systems, and compatible with most popular web hosting software and hardware (Oj, 2020).</p> <p>70% of servers running today are using some form of Linux/Unix. It features a robust feature set, scalability, and a high-performance history. Common distributions of Linux are Ubuntu and CentOS. Most shared hosting plans are user friendly (Horne, 2020).</p> <p>- Unix OS -User and file access controls - Support LDAP and ADP -Docker support</p> <p>Disadvantages are that the expertise level is higher than others and</p>	<p>Server hosting is Windows Server as the operating system and offers Windows-specific technologies.</p> <p>Advantages: Many advancements by Microsoft, well supported. Long term support is provided for all versions. Can be used with a graphical user interface, and system and system applications are well documented (Linux Vs. Windows: A Comparison Of The Best Web Server Solutions, 2021).</p> <p>- Secure -User and file access controls -Support ADP</p> <p>Disadvantages: There is a</p>	<p>Server side available for Mobile as Baas, or Backend as a Service. Frameworks include Database, API's, Storage, Notifications, Authentication.</p> <p>Advantages: Scalability, affordability, hardware-free. Ability to interact with mobile phone features, such as the camera.</p> <p>- iOS developer tools - iOS multi-device support (iPhone, iPad, watch, TV) - Android developer tools</p> <p>Disadvantages are lower flexibility, scalability, data management, and security concerns. Intended for a short-term solution. (Clark, 2020)</p> <p>Android poor multi-device support</p> <p>Licensing costs: Apple Developer license is \$99 per year, \$299 per year for enterprises. Google developer license is \$25 per year. (How Much Does It</p>

<p>-Support LDAP and ADP</p> <p>Disadvantage: The Mac OS Server has a lack of cross-platform client implementation and is bound to Mac OS X clients. (Huges, n.d.)</p> <p>Apple does not offer web hosting; however, the Apache web server can be installed in Amazon Web Services and offers a multitude of capabilities. The Apache web server can cause issues for high volume sites due to its thread-based structure (Domantas, 2021).</p> <p>Licensing costs: Mac server can be added to a Mac OS for \$19.99 (macOSServer, 2021). Apache web server is free and open source.</p> <p>Docker support virtual only</p>	<p>not compatible with Windows OS systems (Oj, 2020).</p> <p>Some third-party applications can only be installed by an administrator, and not all versions come with long-term support. (Linux Vs. Windows: A Comparison Of The Best Web Server Solutions, 2021)</p> <p>Licensing costs:</p> <p>There are no software license fees with Linux, nor its distributions, such as RedHat and CentOS.</p>	<p>challenge with high volumes of activity, frequent rebooting, and the cost tends to be higher than others. (Oj, 2020)</p> <p>Security is an issue as they are prone to errors with integrated interfaces being a potential for attacks (Linux Vs. Windows: A Comparison Of The Best Web Server Solutions, 2021).</p> <p>Licensing costs: \$972 for a standard license, and \$6155 to purchase a data center license. Licenses can also be leased from \$20 a month to \$125 a month, depending on server choice. Some Windows software may also have license costs (Lahn, 2021)</p> <ul style="list-style-type: none"> - Need to use Server version - Limited LDAP - Poor open source server support 	<p>Cost To Make An App In 2021?, 2021)</p>
--	--	---	--

		.	
Client Side			
Mac	Linux	Windows	Mobile Devices
<p>MacOS offers consistent operating system and hardware integration. Advantages are less malware and security issues, incomparable multi-tasking features, and a cost similar to Windows. Time to develop depends on the expertise of the user. Experience level is moderate (Bonheur, 2018).</p> <ul style="list-style-type: none"> -Supports Safari browser - Supports Firefox browser -Supports Chrome browser <p>The development process should include the ability to develop in multiple browsers for all platforms, such as by the use of a browser developer tools or an emulator. Multiple development teams may be needed to ensure full stack web development.</p> <ul style="list-style-type: none"> -Doesn't support Internet Explorer browsers - Small install of client desktops 	<p>Linux clients offer a low cost and offer mainly open-source technology. Time can vary due to the experience level of the user; however, it requires a higher level of expertise. (Oj, 2020)</p> <ul style="list-style-type: none"> -Support Firefox browsers -Supports Chrome browsers <p>The development process should include the ability to develop in multiple browsers for all platforms, such as by the use of a browser developer tools or an emulator. Multiple development teams may be needed to ensure full stack web development.</p> <ul style="list-style-type: none"> - Doesn't support Internet Explorer browsers - Doesn't support Safari browsers - Small install of client desktops 	<p>Windows offers a high degree of specialization and customization. The cost is similar to Mac. The time is moderate due to extra time setting up open-source tools. Moderate expertise, most commonly used OS. (Oj, 2020)</p> <ul style="list-style-type: none"> - Supports Large installed base of desktop clients -Supports Firefox browsers -Supports Chrome browsers -Supports Internet Explorer browsers <p>The development process should include the ability to develop in multiple browsers for all platforms, such as by the use of a browser developer tools or an emulator. Multiple development teams may be needed to ensure full stack web development.</p>	<p>Client-side mobile is varying due to the rapidly changing technology stacks and choice of development platform, such as an emulator. The time to develop is based on the expertise of the development team, but trends toward rapid development. There is a moderate expertise level required. (Clark, 2020)</p> <ul style="list-style-type: none"> - iOS development for Firefox, Chrome, Safari - Android development for Firefox, Chrome <p>An emulator is necessary for the specific version of the mobile app that is being developed and is limited to the mobile app development environments, such as the SDK for the Android applications.</p>

		-Doesn't support Safari browsers	
Development Tools			
Mac	Linux	Windows	Mobile Devices
<p>Common programming languages for Mac are Swift, Python, Java, C, Perl, XCode, JavaScript, HTML5 and CSS.</p> <p>The MacOS can support many popular IDE's open source and free editors such as Sublime, Eclipse, Visual Studio Code, PyCharm, NetBeans, and many others (Kolakowski, 2020).Excels at integration with NodeJS and JavaScript. Enterprise version Eclipse has licensing requirements.</p> <p>Additional advantages:</p> <ul style="list-style-type: none"> - iOS native and Android development supported - High-productivity environment - Large language support including Swift and objective-C - Java support - Unix shell scripting <p>Additional disadvantages:</p> <ul style="list-style-type: none"> - Small pool of developers - Difficult /impossible to test IE browsers 	<p>Common programming languages for Linux are Python, C, C++, Perl, Java.</p> <p>Compatible with many open source and free IDE's such as Atom, Brackets, Sublime, Eclipse, PyCharm, and many others. Enterprise version Eclipse has licensing requirements. (Bolton, 2015)</p> <p>Additional advantages:</p> <ul style="list-style-type: none"> - Large developer pool - New technology first platform - Largest set of IDEs with quality but not always supported - Large language support including Swift and objective-C -.Net support - Java support - Unix shell scripting - Powershell <p>Additional disadvantages:</p> <ul style="list-style-type: none"> -Android dev support only: native 	<p>Common programming languages are C++, HTML5, CSS, Java, JavaScript, Visual Basic, Python, C#, .NET.</p> <p>IDEs are Visual Studio, IntelliJ, Eclipse, WebStorm, PyCharm, NetBeans, and many others. Visual Studio has licensing requirements. (Microsoft, 2017) Enterprise version Eclipse has licensing requirements.</p> <p>Additional advantages:</p> <ul style="list-style-type: none"> -Large set of IDEs with support and quality <p>-Large language support including C#</p> <ul style="list-style-type: none"> - .net support - Powershell -Java support 	<p>Common programming languages for Mobile are Python, Swift, JavaScript, Rust, Scala, Ruby, GoLang, Java, Kotlin.</p> <p>Some IDEs are Visual Studio, Visual Studio Code, Eclipse, Android Studio, IntelliJ, NetBeans, Cordova, and many others. (Blair, 2020)</p>

- Few IDEs, but high quality		Additional disadvantage: - Android dev support only: native	
------------------------------	--	--	--

Recommendation Strategy Comparisons

Operating Systems Architectures

The operating system acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs. It performs basic tasks like file, memory, and process management, input and output, and controlling peripherals. Popular operating systems as detailed in the table above are Linux, Windows, OS X, VMS, AIX, and others (Kumari, 2017).

Some types of Operating Systems are Monolithic, Layered, Microkernel, and Networked and Distributed. The Monolithic operating system architecture is when the entire operating system runs as a single program in kernel mode. Every component of the operating system is contained in the kernel and can communicate directly with each other, with the kernel executing with unrestricted access to the computer system. Monolithic designs can be unwieldy (Kumari, 2017).

Layered Operating System architecture groups components that perform similar functions into layers that can communicate to those above and below it. There is an interface that hides the layer's implementation, with an example being Windows XP. (Kumari, 2017)

Microkernel architecture features process management, file system, networking, and device management outside the kernel with a lower privilege level. Microkernels have a high degree of modularity, which makes them extensible, portable, and scalable, without the microkernel relying on each component to execute. One or more component can fail without causing the operating system to fail. (Kumari, 2017)

Network operating systems enable their processes to access resources such as files that reside on other independent computers or on a network, with the structure often based on the client/server model. The client requests resources, such as the files and processor time, by using

the appropriate network protocol and the servers respond with the appropriate resources.

(Kumari, 2017)

Distributed operating systems are a single operating system that manages resources on one or more computer systems (Kumari, 2017) . A distributed operating system contains multiple components on different machines that are able to coordinate and communicate actions through communications lines called high-speed buses. An important aspect of these systems is the resource sharing, as resources are managed by servers and the clients use the resources (Software Architecture Design- Distributed Architecture, 2021).

File System Components

A file system provides a way to store data as files as well as the functions that can be used to perform operations on the files, such as read, write, open, close, create and delete. A file system is a way to structure a set of data and the metadata that describe the data. In addition to user's files, the file system also contains its own parameters, descriptors, file names, and directory hierarchy and well as extended attributes for storing security information. (The Basics Of File Systems, 2021)

Windows has two major file systems, NTFS, which is the modern version used on the OS by default, and FAT, which inherited from DOS. ReFS was introduced on servers such as Windows Server 2012. NTFS stores files in a master file table with the information about the file, such as size, allocation, and name. (The Basics Of File Systems, 2021)

MacOS has two file systems, the HFS+, and APFS, which is the latest file format on MacOS operating systems running version 10.14 and later. Advanced server products use the Apple Xsan file system. HFS+ uses B-trees for placing and locating files, with volumes divided into sectors that are usually 512 bytes in size, grouped into allocation blocks. Information about free and

used blocks is kept in the Allocation file, file attributes are listed in the Attributes file, and blocks assigned to each file as extends are stored in an Extends File. (The Basics Of File Systems, 2021)

Linux utilizes ext4 file system (Extended File System) as default, and features large filesystem support, improved resistance to fragmentation, higher performance, and improved timestamps (Salter, 2018).

Memory and Storage Management

Main Memory is basically a large array of words or bytes, with each having its own physical address. Main memory provides access from the CPU, and for a program to be executed, it must be in the main memory (personal.kent.edu, n.d.).

Memory management refers to three basic tasks:

1. Keeping track of which part of memory is currently in use and who is using it.
2. To decide which process to load into memory when space is available.
3. Allocate and deallocate memory space as needed. (personal.kent.edu, n.d.)

Memory Management is about understanding what physical address space is available in RAM and “performing memory allocation to place, move, and remove processes from memory address space (Kerner, 2019).“

Memory Management Techniques

Six popular memory management techniques are: Fixed Partitioning, Dynamic Partitioning, Simple Paging, Simple Segmentation, Virtual-Memory Paging and Virtual Memory Segmentation (Alabdulaly, 2016).

Fixed Partitioning is dividing memory into non-overlapping sizes that are fixed, un-moveable, and static. A process may be loaded one at a time into a partition of equal or greater size and is confined to its allocated partition (Alabdulaly, 2016). Dynamic partitioning is defined by partitions that are created dynamically with each process loaded onto a partition that is the same size as the process (Alabdulaly, 2016).

Simple Paging works by writing data to and reading it from secondary storage to be utilized in primary storage. The OS reads data from secondary storage in blocks called pages, all of which have the same size. (Paging, 1999-2021)

Simple Segmentation- This technique is where each job is divided into several segments of different sizes, one for each module that contains pieces that perform related functions, with each segment being a different logical address space of the program (Operating System- Memory Management, 2021).

Virtual Memory Paging- In this storage allocation scheme, secondary memory can be addressed as it were part of main memory. Memory addresses used by a program are called virtual addresses and are mapped into physical addresses in memory (geeksforgeeks.org, 2019).

Virtual Memory Segmentation-Each process is divided into a number of segments, not all of which are resident at any point in time. There is no relationship between physical and physical addresses in segmentation, and a segment table stores information about all segments (Segmentation In Operating Systems, 2019).

Distributed Systems and Networks

A distributed system is a software system in which components located on networked computers communicate and coordinate their actions by passing messages. The components of a

distributed system work together to achieve a common goal, and “consist of a collection of autonomous computers, connected through a network and distributed middleware, which enables computers to coordinate their activities and to share the resources of the system (mjsglobal.com, 2020).”

To a user of the system, it looks like it is one computer when it is really a group of computers. The computers in a distributed system have a “shared state, operate concurrently, and can fail independently without affecting the whole system (Kozlovski, 2018).”

Distributed systems were created in order to fulfill the need of scaling. This gives the computers the ability to scale horizontally, which means to “add more computers instead of upgrading the hardware of one (Kozlovski, 2018).” As a definition of a distributed system, there are “three basic components of a distributed system: concurrency of components, a lack of a global clock, and independent failure of components. A processor has its own private memory and information is exchanged by passing messages between the processor (mjsglobal.com, 2020).”

Some advantages of distributed systems are data sharing, autonomy, and availability. In a distributed system, Some disadvantages are software development cost, a greater potential to bugs, and increased processing overhead.

Distributed System Types:

1. Client Server- the client requests a resource from a server. The server may server multiple clients at the same time. There can be thin-client models and thick-client models.

2. Three-tier- the information about the client is stored in a middle tier in order to simplify development, where presentation, processing, and data management are separated physically.
3. Multi-Tier(N-Tier)-separates an application into layers, developers can create flexible and reusable applications. The most used is three-tier architecture.
4. Peer-to-peer: contains nodes that are equal in data sharing with all tasks divided between them. Nodes interact with each other through a network.
5. Service-Oriented(SOA)-services are components of business functionalities, available to be used with a standard programming interface. Connections between services are conducted by common protocols such as SOAP web service protocol, supports business driven IT where the application consists of services and consumers. (Software Architecture Design- Distributed Architecture, 2021)

Characteristics of distributed systems are heterogeneity, resource sharing, openness, concurrency, scalability, fault tolerance, security, and others (Software Architecture Design- Distributed Architecture, 2021).

Distributed Systems must have a network that connects all of the components together so they can transfer messages and communicate with each other. The network may be connected with an IP address, cables, or a circuit board. The messages are forms of data that are to be shared, such as databases, objects, and files. Important features of the network of a distributed system are the way that messages are communicated: how they are sent, received, acknowledged, and how many times a node may retry on message failure. (Distributed Systems, 2014-2021)

Recommendations

Upon reviewing the Recommendation Strategy Section above, the following recommendations are being presented for The Gaming Room in order to best prepare their game, Draw It Or Lose It, for further distribution:

1. Operating Platform:

An Operating Platform is a “family of servers that includes the hardware and operating system (Platform, 1996-2021).” For example, the Windows Server platform refers to x86 CPUs and some server version of Windows, such as Windows Server 2012. Linux server platforms run on nearly every hardware platform, and are free, open-source operating systems built around the Linux kernel. (Oh, 2018).

For the Draw It Or Lose It Game, the client has suggested an emphasis on security and the ability to further scale the game across multiple clients. In order to provide a suitable suggestion for an operating platform that takes into consideration the various needs of the client, such as scalability and training needs for a web-based application, the recommendation is to use the Linux platform cloud computing such that is provided by the Amazon Web Services platform, Amazon Linux. With the selection of Amazon Web Services, the client will have the ability to further provide their game to a wider selection of clients.

Amazon Web Services will provide a DevOps environment for the game to be hosted and maintained. Amazon Web Services offers a wide availability of choices and features DevOps practices, which would highly benefit the client with the following features:

- Ability to start fast, no setup or software to install
- Full managed services for managing infrastructure
- Scalability for one or thousands of instances

- Programmable options
- Automation
- Security over resources: AWS Identity and Access Management
- Large Partner Ecosystem
- Pay as you go

(Amazon Web Services I. , DevOps and AWS, 2021)

Additional Linux support benefits:

- Server environment, headless/lights out operations
- Multiple architectures: x86, ARM, POWER
- Wide and first support of networking devices/protocols
- Wide support of cloud environments and container technology
- Virtual memory address space
- Shared memory address space
- Wide support of database technologies, such as Oracle, SQLServer, Postgres, MongoDB

2. Operating Systems Architectures:

The Amazon Web Services platform features Amazon Linux2, a flavor of Linux operating system. It provides a “secure, stable, and high-performance execution environment for cloud-based applications (Amazon Web Services I. , 2021).”

This operating system is provided long-term support and access to latest innovations in Linux, with ongoing security and maintenance updates provided. It integrates with the

latest Amazon cloud instances and services and uses the systemd init process to bootstrap user space as well as to manage system processes. (Amazon Linux2, 2021)

The Linux2 is secure by default, with remote access being “limited by using SSH key pairs and disabling remote root logins (Amazon Linux2, 2021)”. The Linux2 reduces the number of non-critical packages installed on an instance, limiting exposure to security vulnerabilities, and features kernel live patching functionality, which limit the need for downtime or rebooting. (Amazon Linux2, 2021)

3. Storage Management:

AWS cloud provides various options for storing, accessing, and backing up web application data and assets. AWS offers the Amazon S3 Simple Storage Service that can be used to access data objects anytime and from anywhere on the internet. Data is stored as objects within resources called buckets, wherein the user can store, read, write and delete from the bucket. AWS offers computes that are specifically designed for games. (Amazon Services, 2021)

The Amazon S3 file storage system is designed for scalability, availability, and durability and can scale up and down to meet fluctuating demands. This file system creates and stores copies of objects across multiple systems and delivers read-after-write consistency (Amazon S3, 2021). Data can be stored in Amazon S3 and secured from unauthorized access with encryption features and access management tools. Amazon Elastic File System, Elastic Block Store, FSx for Windows File Server, FSx for Lustre,

and Amazon Backup are also additional storage and storage management options. (Cloud Storage On AWS, 2021)

4. Memory Management:

AWS offers container instance memory management, which agent uses a Docker function to query available memory of the operating system. There are command line utilities in Linux and windows that can be used to determine total memory. It is possible to reserve system memory for critical system processes. AWS also offers a CodeGuru profiler that will help to understand memory usage for Java applications (Amazon Services, 2021).

“The Linux OS serves as a layer of abstraction between applications and physical memory. This layer divides physical RAM into pages, or the smallest units of data an OS can use. When Linux uses RAM, it creates a virtual memory layer and then assigns processes to virtual memory (How Does Linux Handle Ram?, 2018)”. Virtual memory is a combination of both RAM and swap space. For Linux, a “swap space” is a section of the hard drive designated as available for use in case usable RAM runs out. This extra abstraction layer ensures that each running process doesn't overlap and try to use memory already being used by another process. Virtual memory can be expanded beyond the physical RAM capacity (How Does Linux Handle Ram?, 2018).

Linux uses the command “mmap” to map files, referred to as a “memory mapped file”. If a memory page file doesn't have any file associated with it, it's referred to as anonymous memory and allocated using the "malloc" function. With the way that file mapped memory and anonymous memory are allocated, the operating system can have

processes using the same files working with the same virtual memory page thus using memory more efficiently (How Does Linux Handle Ram?, 2018).

In Linux, if a process is occupying memory that's needed for another one, the OS uses the OOM (out of memory) killer. This utility chooses a process and reallocates its memory pages to other processes. In Linux, this OOM killer is enabled by default (How Does Linux Handle Ram?, 2018).

Amazon EC2 instances use swap space as a short-term replacement for physical RAM if the physical RAM is already in use. RAM that isn't in use or not actively needed is temporarily paged to a swap file. Swap space can also be used on a partition. Linux divides physical RAM into chunks of memory called pages (How do I allocate memory to work as swap space in an Amazon EC2 instance by using a swap file?, 2021).

Swap space is the process “whereby a page of memory is copied to the preconfigured space on the hard disk to free up that page of memory. The combined sizes of the physical memory and the swap space is the amount of virtual memory available (Sims, 2007)”.

5. Distributed Systems and Networks:

Cloud computing offers the flexibility to design with the client's application needs in mind. The responsibility for connectivity and outage management falls on into the hands of the provider.

The distributed network suggested is the Client-Server Model distributed network system, albeit the client will be a virtual client, and the server a virtual server. This Model is a process consists of three parts: the client, the server, and the network. The client is an application that is running the front-end implementations and will request processes

from the server and then display the data that the server returns. The server is a virtual machine that handles the functions of returning data to the client and handles the functions for shared data access. The server is referred to as the back end, and/or the server process that runs on a server machine/virtual machine.

The network is what delivers the messages back and forth from the client to the server and vice versa, allowing remote data access (Understanding Distributed Systems, 1996). The application programming interface(API) defines and determines how the client-server architecture processes service requests between a client and a server over a network connection. The sample Java API code base has been constructed for the web application and will provide the means to distribute the game over numerous clients.

AWS features build your own or use managed services to create customized server infrastructure in the cloud or select managed services in order to scale and save time. Reliable network bandwidth should be in place for the developer, and it is the responsibility of AWS to provide sufficient networking and compute capacity. The client will be free to change resource size and allocations on demand. (Amazon Services, 2021)

As a part of the distributed system, AWS features a key-value document database called DynamoDB, which is highly used for game applications. It features “fully-managed, multi-region, multi-active, built-in security, backup and restore, and in-memory caching.” DynamoDB is serverless and scalable to adjust up and down as needed, with availability and fault-tolerance built in. This database is widely used in serverless web applications and, mobile, backends, and microservices. (DynamoDB, 2021)

6. Security:

AWS has high standards for privacy and data security. It features a secure, global infrastructure where the client will always own their data with the ability to encrypt it, move it, and manage the retention. All data flowing across Amazon data centers is automatically encrypted at the physical layer before it leaves secured facilities. Additional encryption layers exist as well. AWS also offers the largest ecosystem of security partners and the ability to inherit the most comprehensive security and compliance controls. (Amazon Services, 2021)

In addition to protecting the data that is stored for the application, user authentication and authorization is a priority for the game. The authentication and authorization concept that was modeled in the sample code is an example of how to ensure that the user is authenticated and to assign them roles in order to give users certain capabilities (Getting Started With User Pools, 2021).

AWS also offers secure and scalable SAML authentication as a process of verifying the users, as well as IAM as a feature of the AWS account. IAM is used to create and manage users and groups to provide various accesses to AWS resources. Extensive tutorials and information about these authentication services is provided for free at AWS (AWS Identity And Access Management(IAM), 2021).

References:

Alabdulaly, W. (2016). Memory Management Techniques and Process Scheduling. International Journal of Scientific & Engineering Research, Volume 7, Issue 4, 1182-1184.

Amazon Linux2. (2021). Retrieved April 2021, from aws.amazon.com:

<https://aws.amazon.com/amazon-linux-2/>

Amazon S3. (2021). Retrieved April 2021, from aws.amazon.com: <https://aws.amazon.com/s3/>

Amazon Services, I. (2021). AWS Compute for Games. Retrieved March 2021, from

aws.amazon.com: <https://aws.amazon.com/gametech/compute/>

Amazon Web Services, I. (2021). AWS Cloud Security. Retrieved March 2021, from

aws.amazon.com: <https://aws.amazon.com/security/>

Amazon Web Services, I. (2021). Container Instance Memory Management. Retrieved March 2021, from docs.aws.amazon.com:

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/memory-management.html>

Amazon Web Services, I. (2021). DevOps and AWS. Retrieved March 2021, from

aws.amazon.com: https://aws.amazon.com/devops/?nc2=h_ql_sol_use_dops

Apple. (n.d.). Xgrid. Retrieved March 2021, from [apple.com](https://www.apple.com):

https://www.apple.com/server/docs/Xgrid_TB_v10.4.pdf

AWS Identity And Access Management(IAM). (2021). Retrieved April 2021, from

aws.amazon.com: <https://aws.amazon.com/iam/>

Blair, I. (2020). 14 Programming Languages For Mobile App Development. Retrieved March 2021, from buildfire.com: <https://buildfire.com/programming-languages-for-mobile-app-development/>

Bolton, D. (2015, March 09). Best Programming Languages For Linux Devs. Retrieved March 2021, from insights.dice.com: <https://insights.dice.com/2015/03/09/best-programming-languages-linux-devs/>

Bonheur, K. (2018, April 23). Advantages and Disadvantages of MacOS. Retrieved March 2021, from profolus.com: <https://www.profolus.com/topics/advantages-and-disadvantages-of-macos/>

Clark, J. (2020). Top 10 Mobile App Backend Hosting Providers. Retrieved March 2021, from back4app.com: <https://blog.back4app.com/mobile-app-backend-hosting/>

Cloud Storage On AWS. (2021). Retrieved April 2021, from aws.amazon.com: <https://aws.amazon.com/products/storage/>

Distributed System. (2020). Retrieved April 2021, from mjsglobal.com: <https://mjsglobal.com/distributed-system/>

Distributed Systems. (2014-2021). Retrieved April 2021, from confluent.io: <https://www.confluent.io/learn/distributed-systems/>

Domantas, G. (2021, March 09). What Is Apache Webserver? Retrieved April 2021, from hostinger.com: <https://www.hostinger.com/tutorials/what-is-apache>

DynamoDB. (2021). Retrieved April 2021, from aws.amazon.com: <https://aws.amazon.com/dynamodb/?c=db&sec=srv>

geeksforgeeks.org. (2019, August 19). Retrieved April 2021, from Virtual Memory In Operating Systems: <https://www.geeksforgeeks.org/virtual-memory-in-operating-system/>

Getting Started With User Pools. (2021). Retrieved April 2021, from docs.aws.amazon.com:

<https://docs.aws.amazon.com/cognito/latest/developerguide/getting-started-with-cognito-user-pools.html>

Horne, K. (2020, July 03). Operating Systems and Web Hosting: Read Here Before Making Your Choices. Retrieved April 2021, from whoishostingthis.com:

<https://www.whoishostingthis.com/compare/operating-systems/>

How do I allocate memory to work as swap space in an Amazon EC2 instance by using a swap file? (2021, March 09). Retrieved April 2021, from aws.amazon.com:

<https://aws.amazon.com/premiumsupport/knowledge-center/ec2-memory-swap-file/>

How Does Linux Handle Ram? (2018, April 09). Retrieved March 2021, from Serversuit.com:

<https://serversuit.com/community/technical-tips/view/how-does-linux-handle-ram.html#:~:text=When%20Linux%20uses%20system%20RAM,case%20usable%20RAM%20runs%20out.>

How Much Does It Cost To Make An App In 2021? (2021). Retrieved from velvetech.com:

<https://www.velvetech.com/blog/how-much-mobile-app-cost/>

Huges, B. (n.d.). Building Computational Grids with Apple's Xgrid Middleware. Retrieved March 2020, from dl.acm.org: <https://dl.acm.org/doi/pdf/10.5555/1151828.1151835>

Kerner, S.-M. (2019, July 19). Memory Management: How It Works & Why You Need It? Retrieved April 2021, from enterprisestorageforum.com:

<https://www.enterprisestorageforum.com/hardware/memory-management/>

Kolakowski, N. (2020, 12 01). 7 Programming Languages Popular At Apple That Could Land You A Job. Retrieved March 2021, from insights.dice.com:

<https://insights.dice.com/2020/12/01/7-programming-languages-popular-at-apple-that-could-land-you-a-job/>

Kozlovski, S. (2018, April 27). A Thorough Introduction To Distributed Systems. Retrieved April 2021, from freecodecamp.org: <https://www.freecodecamp.org/news/a-thorough-introduction-to-distributed-systems-3b91562c9b3c/>

Kumari, S. (2017, August 22). Architecture Of Operating Systems. Retrieved April 2021, from slideshare.net/SupriyaKumari54:
<https://www.slideshare.net/SupriyaKumari54/architecture-of-operating-system>

Lahn, M. (2021, January 11). How Much Do Windows Servers Cost? Retrieved April 2021, from servermania.com: <https://www.servermania.com/kb/articles/how-much-does-a-windows-server-cost/>

Linux Vs. Windows: A Comparison Of The Best Web Server Solutions. (2021, 01 14). Retrieved April 2021, from ionos.com: <https://www.ionos.com/digitalguide/server/know-how/linux-vs-windows-the-big-server-check/>

macOSServer. (2021). Retrieved April 2021, from apple.com:
<https://www.apple.com/macos/server/#:~:text=Add%20macOS%20Server%20to%20your,App%20Store%20for%20just%20%2419.99.>

Microsoft. (2017, 02 08). Getting Started: Choosing A Programming Language. Retrieved March 2021, from docs.microsoft.com: <https://docs.microsoft.com/en-us/windows/uwp/porting/getting-started-choosing-a-programming-language>

mjsglobal.com. (2020). Retrieved April 2021, from Distributed System:
<https://mjsglobal.com/distributed-system/>

Oh, D. (2018, May 28). What Is A Linux Server And Why Does Your Buisness Need One?

Retrieved April 2021, from opensource.com: <https://opensource.com/article/18/5/what-linux-server>

Oj, M. (2020, April 2). Windows vs Linux Hosting, The Advantages and Disadvantages.

Retrieved March 2021, from vpcart.com: <https://www.vpcart.com/blog/windows-vs-linux-hosting-the-advantages-and-disadvantages/>

Operating System- Memory Managment. (2021). Retrieved April 2021, from tutorialspoint.com:

https://www.tutorialspoint.com/operating_system/os_memory_management.htm#:~:text=Segmentation%20is%20a%20memory%20management,address%20space%20of%20the%20program.

Paging. (1999-2021). Retrieved April 2021, from whatis.techtarget.com:

<https://whatis.techtarget.com/definition/paging#:~:text=Paging%20is%20a%20function%20of,storage%20to%20the%20primary%20storage.&text=It%20is%20typically%20stored%20in,for%20longer%20periods%20of%20time.>

personal.kent.edu. (n.d.). Retrieved April 2020, from System Components:

<http://personal.kent.edu/~rmuhamma/OpSystems/Myos/sysComponent.htm>

Platform. (1996-2021). Retrieved April 2021, from pcmag.com:

<https://www.pcmag.com/encyclopedia/term/platform#:~:text=An%20Operating%20System%20Platform,on%20x86%20machines%20for%20decades.>

Salter, J. (2018, April 02). Understanding Linux Filesystems: ext4 and Beyond. Retrieved April

2021, from opensource.com: <https://opensource.com/article/18/4/ext4-file-system>

Segmentation In Operating Systems. (2019, August 16). Retrieved April 2021, from

[geeksforgeeks.org: https://www.geeksforgeeks.org/segmentation-in-operating-system/](https://www.geeksforgeeks.org/segmentation-in-operating-system/)

Sims, G. (2007, September 07). All About Linux Swap Space. Retrieved April 2021, from
linux.com: <https://www.linux.com/news/all-about-linux-swap-space/>

Software Architecture Design- Distributed Architecture. (2021). Retrieved April 2021, from
tutorialspoint.com:
https://www.tutorialspoint.com/software_architecture_design/distributed_architecture.htm

The Basics Of File Systems. (2021, March 23). Retrieved April 2021, from ufsexplorer.com:
<https://www.ufsexplorer.com/articles/file-systems-basics.php>

Understanding Distributed Systems. (1996). Retrieved April 2021, from docs.oracle.com:
https://docs.oracle.com/cd/A57673_01/DOC/server/doc/SD173/ch1.htm