Table of Contents

CHAPTER 1: INTRODUCTION

1.1 Background Study

Traditional learning, which means face-to-face classes at school or college, has several problems. Students usually need to travel to the classroom, which can be difficult for those who live far away or have transport issues. The learning schedule is fixed, so students cannot study at their own pace or time. Some students also have other responsibilities like part-time jobs, taking care of family, or housework, which makes it hard to attend every class. Educational institutions also need to spend a lot on things like printing learning materials, buying whiteboards, and managing classrooms. Because of these challenges, the rising demand for online learning prefers a more flexible and easier way to learn, particularly since the COVID-19 pandemic made educational institutions reconsider their methods [1] (*refer to Appendix 2.1*). Previously, many academic institutions used outdated Learning Management Systems (LMS) that operated independently on servers [2]. These were complex systems that made it difficult for students in different time zones to access resources efficiently, were costly to operate, and were unable to handle many users at once.

Cloud computing refers to the delivery of computing services, including servers, storage, databases, networking, software, and analytics over the Internet, particularly Amazon Web Services (AWS), which offer solutions to address these challenges. By leveraging cloud-based platforms, educational institutions can provide students with 24/7 access to learning materials from anywhere and anytime that facilitates flexible learning schedules. Cloud services also support scalability, allowing institutions to accommodate varying numbers of users without significant infrastructure changes [3]. Additionally, cloud computing reduces the need for physical storage and maintenance, leading to cost savings and increased efficiency. Despite all this, AWS enables schools to pay only for what they use, scale up or down in response to demand, and deliver material globally without delays.

Considering AWS is widely used in academic contexts and has a competitive advantage over competitors like Google Cloud Platform and Microsoft Azure, it has been chosen for this study. Services like CloudFront provide seamless content delivery even in areas with low bandwidth, Amazon S3 safely saves educational resources, and Amazon EC2 makes it possible to create scalable virtual classrooms. Furthermore, AWS Educate offers colleges free resources, which makes it perfect for universities on a tight budget. Although Google Cloud and Azure both have special characteristics, AWS is a better option for e-learning platforms due to its global server network, education-specific tools (such as pre-made templates for tests), and reliability.

Making the switch to cloud-based learning is not without challenges, though. Many institutions lack the technical know-how needed to integrate historical systems with AWS products like AWS Lambda (for task automation) or Amazon RDS (for database conversion). Security is still an issue; proper setup of tools like IAM is necessary to guarantee compliance with laws like data protection while storing student data on AWS. Despite these challenges, cloud platforms reinforce their position as the educational future by relieving teachers of IT responsibilities and giving students constant access to materials.

Despite the challenges faced when switching to the cloud, cloud-based learning systems offer distinct advantages to various stakeholders in the educational ecosystem. For students, these systems provide flexibility, personalized learning paths, and access to a wide range of resources. Educators benefit from streamlined content delivery, real-time feedback mechanisms, and data analytics to monitor student progress. Institutions can reduce operational costs, scale services according to demand, and enhance their technological infrastructure without significant capital investment. In response to the identified challenges and opportunities, this technical solution proposes the development of an online learning platform hosted on AWS. By leveraging AWS, our ULearnly online learning website aims to deliver an accessible, efficient, and user-friendly learning experience that addresses the limitations of traditional educational models and meets the evolving needs of learners and educators alike.

1.2 Problem Statement

1. Scalability and Performance Bottlenecks: Traditional e-learning platforms often struggle to manage sudden spikes in user traffic, during peak times such as exams or live sessions. This may leads to latency, downtime or even system crashes, affecting overall learning experience. There is a need for a scalable infrastructure that can adapt to varying user loads.

2. Cost Management and Resource Efficiency: While cloud infrastructure reduces upfront costs, institutions risk overspending due to inefficient resource provisioning. This can become a long-term financial burden. Efficient cost control mechanisms are essential to maintain low operational costs.

3. Data Security and Compliance: Storing sensitive student data on the cloud about unauthorized access, data breaches, and regulatory compliance (e.g., PDPA). A secure environment with strong access control and data privacy is required.

4. Integration Complexity: Many educational institutions still rely on legacy systems, such as on-premise databases or traditional content management platforms. Integrating these with AWS-native tools (e.g., migrating databases to Amazon RDS) often requires advanced technical knowledge, which may not be available in all organizations. This creates a barrier for institutions that lack dedicated IT teams or cloud migration expertise.

5. Global Accessibility Disparities: Learners in regions with limited Internet bandwidth face challenges access resource-heavy content like high-definition videos or real-time features. This digital divide leads to unequal learning experiences. There is a need for a platform that can deliver optimized content suitable for low-bandwidth environments to ensure inclusive access.

1.3 Objectives

1. To analyze and evaluate the use of AWS as a scalable, secure, and cost-effective solution for hosting an online learning platform.

2. To design and deploy a cloud-based online learning platform using AWS services such as EC2, RDS, EFS, and Auto Scaling to ensure high availability, performance, and centralized resource access.

3. To provide educators and learners with a reliable, accessible, and interactive digital learning environment that supports consistent performance and data availability regardless of user location or internet speed.

CHAPTER 2: LITERATURE REVIEW

2.1 Overview of AWS for Website Hosting

This section provides a foundation for understanding why AWS is widely adopted for website hosting by outlining its background and the key advantages that make it a preferred choice for businesses and institutions alike.

History of AWS

AWS was officially launched on March 14, 2006, by Amazon.com under the leadership of Andy Jassy, who later became the CEO of Amazon [4]. The original idea for AWS emerged in 2003 when Amazon engineers Chris Pinkham and Benjamin Black proposed building standardized, automated infrastructure using web services. Along with Christopher Brown and Willem van Biljon, the AWS team began developing the early services in Cape Town, South Africa [4]. The launch included foundational services like Amazon S3 and EC2, marking the beginning of modern cloud computing (*refer to Appendix 2.2)*. Since then, AWS has grown into the world's largest cloud provider, serving millions of active customers globally.

A critical part of AWS's reliability and performance lies in its global infrastructure. This infrastructure is made up of four core components: Regions, Availability Zones, Edge Locations, and Regional Edge Caches. As updated in 2024, AWS operates in 36 geographic regions, with plans to launch more, and includes over 114 Availability Zones (AZs) that allow fault-isolated clusters of data centres within each region. Additionally, AWS supports content delivery and low-latency services via over 400 Edge Locations and 13 Regional Edge Caches, ensuring rapid content access and resilience around the world [5] (*refer to Appendix 2.3*). These components work together to deliver highly available, secure, and scalable applications to users anywhere across the globe.

Prominent Companies Utilizing AWS

AWS has become the backbone for numerous well-known companies across various industries, helping them to leverage its scalability, reliability, and global reach. These examples underscore AWS's capability to support diverse and demanding workloads, making it a preferred choice for organizations aiming to deliver robust and scalable web-based services.

1. **NASA:** The U.S. space agency employs Amazon Web Services (AWS) to power its Image and Video Library, providing public access to over 140,000 high-quality photos, videos, and audio files. By leveraging AWS services such as Amazon S3 for scalable storage and Amazon CloudFront for fast global content delivery, NASA ensures that users can search and stream media efficiently from anywhere in the world. The cloud-based architecture eliminates the need for on-premises infrastructure, allowing NASA to manage large amounts of multimedia data with minimal operational burden. With AWS, NASA delivers a highly available, responsive, and cost-effective platform that supports its mission of public engagement and open access to scientific data [6].
2. **Netflix:** As a leading streaming service, Netflix utilizes AWS to manage and analyze billions of messages daily across over 100,000 application instances. This enables real-time optimization of user experience, cost reduction, and improved application resilience [7].
3. **Airbnb:** Airbnb leveraged Amazon Web Services (AWS) to overcome early infrastructure limitations and scale rapidly as user demand grew. By migrating to AWS, Airbnb gained the ability to deploy servers on demand, store large volumes of user data with Amazon S3, and ensure high application availability using Elastic Load Balancing and EC2 instances. The adoption of Amazon RDS allowed the company to automate database management, including replication and scaling, with minimal downtime. Additionally, AWS services like CloudWatch and EMR enabled real-time monitoring and efficient data processing. Overall, AWS provided Airbnb with a cost-effective, flexible, and scalable infrastructure that supported its global growth and operational agility. [8].

Benefits of using AWS for Website Hosting

The advantages of utilizing AWS for website hosting are manifold. Its **scalability** allows businesses to easily adjust computing resources based on traffic demands to guarantee that websites remain responsive during traffic spikes [9]. The **elasticity** of AWS services further enables seamless resource allocation, automatically scaling up or down based on workload fluctuations [9]. **Security** is built into the AWS platform through well-structured, multi-layered protections, encryption, and compliance certifications to help mitigate risks and protect sensitive data [9].

Additionally, AWS boasts a **rich ecosystem** of services, including EC2 for computing power, Lambda for serverless execution, RDS for managed databases, and CloudFront for content delivery while providing businesses with comprehensive tools to optimize performance and efficiency [9]. Its **global infrastructure** spanning multiple regions and availability zones guarantees minimal latency and high availability for users worldwide [9]. Furthermore, AWS provides **cost-efficiency** through a pay-as-you-go pricing model that allows organizations to optimize spending without requiring significant upfront investments [9]. Collectively, these features position AWS as the cornerstone for digital transformation in today's competitive market.

To sum up, AWS offers a comprehensive and flexible ecosystem for website hosting that spans compute, storage, databases, networking, security, and deployment tools. Its diverse services and resilient infrastructure not only simplify the process of deploying and managing websites but also provide enterprises with the scalability, security, and cost-efficiency necessary for sustained growth. As organizations continue to adopt cloud-based solutions, AWS remains at the forefront of innovation, continually evolving to meet the complex needs of the digital age.

2.2 Comparative Analysis of AWS and Other Cloud Service Providers

This section presents a comparative analysis of AWS and its major competitors, Google Cloud Platform (GCP), Microsoft Azure, and IBM Cloud, based on several critical aspects including pricing, security, developer tools, and support for education. The comparison helps highlight AWS's strengths, limitations, and overall suitability as a cloud platform for hosting online learning website solutions.

Google Cloud Platform (GCP)

Google Cloud Platform (GCP) is a suite of cloud computing services developed by Google, running on the same secure and high-performance infrastructure used for Google's global products such as Google Search, Gmail, and YouTube. It is designed to deliver scalable computing, advanced data analytics, and AI services across a global network of data centres, currently spanning over 42 regions and 100-plus zones worldwide [10]. GCP is particularly known for its strength in data analytics, machine learning, and containerized applications. It is the birthplace of Kubernetes, and its flagship services like BigQuery (a serverless data warehouse) and Vertex AI distinguish it from other cloud providers. GCP also provides services like AWS, such as Compute Engine (equivalent to AWS EC2), Cloud Storage (similar to Amazon S3), and Cloud Functions (like AWS Lambda) [10]. However, it differs in its native integration with Google Workspace, and its AI or Machine Learning ecosystem is more tightly coupled with Google’s research advancements. GCP is also used by many major companies and institutions, including Spotify (for music streaming infrastructure), Snap Inc. (Snapchat backend services), and Twitter (for real-time analytics and content delivery). Its focus on developer-friendly interfaces, performance, and innovation makes it a strong competitor in the cloud services landscape.

Microsoft Azure

Microsoft Azure is a comprehensive cloud computing platform launched by Microsoft in 2010, offering a wide range of services through a global network of data centres. As updated in 2024, Azure operates in over 60 regions with two hundred plus availability zones, making it one of the largest and most globally distributed cloud infrastructures [11]. Microsoft Azure is widely adopted for enterprise solutions, particularly among organizations already using Microsoft products such as Windows Server, SQL Server, and Active Directory. It supports various services that mirror AWS offerings, including Azure Virtual Machines (like AWS EC2), Azure Blob Storage (like Amazon S3), and Azure Functions (comparable to AWS Lambda) [11]. However, Azure's tight integration with Microsoft 365, Dynamics, and GitHub gives it an edge in hybrid cloud deployments and enterprise productivity suites. Many large-scale enterprises trust Azure for their cloud needs, including Coca-Cola (global operations and customer engagement platforms), Volkswagen Group (vehicle software development), and Adobe (cloud-based creative software delivery). Its enterprise-grade support, hybrid compatibility, and developer ecosystem make it a top choice for large organizations transitioning to the cloud.

IBM Cloud

IBM Cloud is a cloud computing platform developed by IBM, offering a combination of infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). IBM Cloud places a strong emphasis on hybrid cloud, AI integration (via Watson), and enterprise-grade security. It operates in over 60 cloud data centres globally, with a focus on industries that require strict compliance and data governance, such as healthcare, banking, and government [12]. IBM Cloud provides many services similar to AWS, such as Virtual Servers (similar to EC2), Cloud Object Storage (like Amazon S3), and IBM Cloud Functions (comparable to AWS Lambda) [12]. However, IBM differentiates itself through services like Watson AI (natural language processing and cognitive computing) and Red Hat OpenShift integration for containerized application deployment. Its hybrid approach via IBM Cloud Satellite also makes it uniquely positioned for organizations that need to run cloud workloads across on-premises and edge environments. Notable organizations using IBM Cloud include Maersk (for secure global supply chain management), American Airlines (for hybrid cloud solutions in flight operations), and The Weather Company (to manage vast amounts of meteorological data). IBM's focus on compliance, data privacy, and AI positions it as a strong choice for businesses requiring controlled and secure environments.

Comparative analysis between different cloud services provider:

Aspect	AWS	GCP	Microsoft Azure	IBM Cloud
Global infrastructure	36 regions 114 AZ	42 regions 100+ AZ	60+ regions 200+ AZ	No exact regions 42 AZ
Performance and Reliability	High reliable, multi-AZ redundancy	High-speed global network, reliable structure	Highly redundant, enterprise-grade SLA	Reliable for hybrid workloads, smaller global footprint
Competitive advantage	Broadest service portfolio, mature ecosystem	Best for analytics and AI or machine learning workloads	Best for enterprises using Microsoft tools	Strong in hybrid cloud, AI (Watson), and compliance
Pricing				
Pricing model	Pay-as-you-go, reserved, spot instances	Pay-as-you-go, sustained-use discounts	Pay-as-you-go, reserved, spot, enterprise licensing	Hourly / monthly pricing, enterprise agreements

Free tier	12-month free tier, limited service always free	Free tier with $300 credits for 90 days	$200 credit for 30 days, 12-month free tier	Life plan available; no full unified free tier
Security				
Encryption	Data encryption in transit and at rest	End-to-end encryption	Encryption in transit and at rest	Strong encryption, private networking
Identity and Access Management	IAM role for access control	Cloud IAM with resource-level policies	Azure Active Directory Integration	IAM with granular policies
Compliance certification	ISO, SOC, HIPAA, GDPR	ISO, SOC, GDPR, FedRAMP	ISO, HIPAA, GDPR, FedRAMP	ISO, GDPR, HIPAA, industry-specific standards
Security operation	CloudWatch, GuardDuty, Shield, WAF 24/7 security monitoring	Security Command Center, threat detection	Azure Security Centre, Sentinel	IBM Security, Qadar integration
Scalability				
Elastic compute	EC2 instances (scalable compute), Lambda (serverless), auto-scaling	Compute Engine, App Engine, Cloud Functions, Autoscaler	Virtual Machines VM Scale Sets, Azure Functions (serverless)	Virtual Servers, Red Hat OpenShift for containers, Bare Metal Servers
Load balancing	ELB	Cloud Load Balancing	Azure Load Balances	IBM Load Balancer
On-demand resources	EC2 Spot Instances, On-Demand Instances	Pay-as-you-go VMs	Pay-as-you-go VMs, Reserved Instances	Hourly and monthly virtual servers, reserved capacity options
User-friendliness				
Management console	AWS management console	Google Cloud Console	Azure Portal	-
Ease of integration and user experience	Well-documented, robust console, moderate learning curve, requires training	Clean user interface, great documentation, developer-friendly	Tight integration with Microsoft ecosystem, good documentation with Microsoft support	Basic interface; enterprise-focused, less intuitive
Customer support and global reach	Global availability, strong community and partner network	Global presence, strong in data and AI sectors	Enterprise-focused support, strong global reach	Limited reach compared to other 3, focused support

Based on the comparative analysis, AWS provides the most comprehensive global reach with the largest number of availability zones worldwide, ensuring low-latency access to users globally, which is especially important for online learning websites with international learners. AWS also offers superior scalability with services like EC2 instances and S3 storage, ensuring excellent performance even during traffic spikes. Therefore, **AWS emerges as the most suitable choice for hosting an online learning website.** It offers the most mature, globally distributed infrastructure, extensive education-friendly free tier, robust security and compliance, and an unmatched range of services. Its scalability through services like EC2, ELB, and Auto Scaling ensures reliable performance during peak usage, and its user support ecosystem is vast. While Google Cloud excels in AI or machine learning and Microsoft Azure integrates well with enterprise tools, AWS strikes the best balance between scalability, reliability, flexibility, and education readiness—making it ideal for supporting diverse learners and large-scale deployments in the education sector.

2.3 Case Studies on AWS Implementation for Website Solutions

AWS has supported numerous businesses in creating resilient and scalable website solutions. This section highlights real-world implementations of AWS by well-known organizations to demonstrate its effectiveness in hosting high-performance, scalable, and globally accessible websites while discussing implementation challenges. Through case studies, the section illustrates how AWS supports diverse use cases, from dynamic marketplaces and streaming services to public media archives, reinforcing its reliability and flexibility across industries.

Real-World Examples of Websites Built on AWS

Numerous high-traffic websites leverage AWS to meet their demanding operational needs. For instance, **global e-commerce platforms** like Shopify and Shopline utilize AWS to manage peak traffic during seasonal sales, while popular **social media sites** such as Facebook, Twitter (now known as X), and Reddit benefit from its low-latency content delivery and real-time data processing [13][14][15][16][17]. These examples illustrate how AWS enables companies to not only deploy robust solutions but also innovate continuously in a highly competitive digital landscape.

Challenges Faced and Solutions Provided by AWS

Implementing a comprehensive cloud infrastructure is not without its challenges. One common issue is **managing sudden surges in traffic**. AWS addresses this through its Elastic Compute services such as EC2 and auto-scaling mechanisms, which dynamically allocate resources to meet demand [18]. Additionally, **ensuring data security and regulatory compliance** poses significant hurdles. AWS responds with its suite of security tools like IAM, Shield, and CloudTrail, offering businesses a multi-layered defense against threats and ensuring adherence to stringent compliance standards [19]. Furthermore, **the deployment complexities across diverse environments** are mitigated by DevOps tools like CodePipeline and CloudFormation, which streamline continuous integration and delivery processes [20].

Successful Implementations in Diverse Sectors

E-Commerce

In the e-commerce sector, AWS's powerful infrastructure supports not only high transaction volumes but also complex inventory management and personalization algorithms. The integration of services such as RDS for relational databases and DynamoDB for NoSQL operations ensures that data is readily available for analytics and customer insights, ultimately enhancing better business decisions [21].

Social Media

Social media platforms demand real-time data processing and efficient content delivery. AWS's networking services such as VPC and Route 53 operate alongside its global content delivery network (CDN) to minimize latency [5][16][17][22]. This smooth experience is essential for user engagement and retention and makes AWS a preferred option for platforms with millions of active users.

Blogs and Content-Driven Sites

For blogs and content-heavy websites, scalability and reliability are paramount. AWS S3 offers a reliable solution for storing and delivering static content while keeping websites accessible during traffic spikes [23]. Additionally, the integration of AWS Lambda for serverless computing allows these platforms to execute backend tasks efficiently without incurring the overhead of traditional server management.

Learning Platforms

Online learning platforms require a combination of robust database management and flexible computing power to support interactive features and real-time collaboration. AWS services like Aurora and Elastic Beanstalk enable smooth scaling of these platforms to ensure learners have uninterrupted access to educational resources during peak usage periods [18].

To sum up, the real-world case studies underscore AWS's capability to provide tailored solutions across various industries. From managing high-traffic e-commerce websites to supporting interactive learning environments, AWS addresses complex challenges with innovative services that emphasize scalability, security, and cost-efficiency. These successful implementations show how AWS supports various website solutions while driving digital transformation to keep businesses agile and competitive in a constantly evolving technological landscape.

2.4 Advanced Considerations of AWS in Web Infrastructure

In addition to its robust suite of core services, AWS offers advanced features that address critical considerations for modern enterprises. These aspects include complete security and compliance measures, flexible pricing models, and cost-saving strategies along with a strong focus on sustainability and green computing. This section explores these advanced considerations and explains how AWS helps organizations navigate complex challenges while optimizing operational efficiency.

Security and Compliance Considerations

Ensuring data security and regulatory compliance is a primary concern for any organization operating in the digital space. AWS addresses these needs by providing a range of tools and policies that safeguard data integrity and privacy. Central to this is AWS's data protection strategy, which adheres to strict privacy policies aligned with global regulations. Advanced encryption mechanisms protect data both at rest and in transit, ensuring that sensitive information remains secure [24]. Additionally, AWS provides granular access control through Identity and Access Management (IAM), enabling organizations to define precise permissions for users and services. This not only minimizes potential vulnerabilities but also simplifies compliance with regulatory standards such as GDPR, HIPAA, and PCI DSS [25].

Pricing Models and Cost-Saving Strategies

AWS employs a flexible, pay-as-you-go pricing model that allows businesses to optimize costs by paying only for the resources they consume [26]. Alongside this, AWS offers Reserved Instances and Spot Instances which provide additional opportunities for cost savings. Reserved Instances offer long-term commitments at lower rates while Spot Instances let users bid on unused capacity which

often leads to substantial savings for non-critical workloads. Furthermore, AWS Free Tier enables new users to explore services with no upfront cost, which is beneficial during initial deployments. Auto Scaling allows resource adjustment based on real-time demand [26], further supporting efficient spending. To help organizations manage expenses, AWS offers tools like Cost Explorer, Budgets, and Savings Plans, which enable accurate forecasting, real-time monitoring, and strategic resource allocation [26].

Sustainability and Green Computing

In today's environmentally conscious market, AWS's commitment to sustainability is a key differentiator. AWS has invested in carbon-neutral data centers that run on renewable energy sources showing its commitment to reducing environmental impact. Energy-efficient computing is further enhanced through innovations like the AWS Graviton processors, which provide high performance with reduced power consumption [27]. Many businesses have embraced AWS's sustainability initiatives by integrating these eco-friendly technologies into their infrastructure. Case studies from various industries highlight how companies not only benefit from lower operational costs but also contribute to broader corporate social responsibility goals by reducing their carbon footprints [27].

To sum up, advanced considerations in AWS go far beyond basic cloud hosting, addressing essential areas such as security, cost management, and environmental sustainability. With robust security and compliance features, flexible pricing options, and innovative green computing solutions, AWS equips organizations with the tools they need to achieve operational excellence and strategic growth. These advanced offerings underscore AWS's role as a comprehensive cloud platform that not only meets current business needs but also anticipates future challenges in a rapidly evolving technological landscape.

CHAPTER 3: METHODOLOGY

3.1 Category of AWS services

Compute Services

AWS offers various computing options to meet different operational needs. **EC2** is a popular service that provides scalable computing capacity in the cloud and enables businesses to adjust resources based on demand [18]. **Lambda** takes a serverless approach that enables developers to run code without provisioning or managing servers and is ideal for event-driven applications [18]. **Elastic Beanstalk** simplifies application deployment by automatically handling capacity provisioning, load balancing, scaling, and application health monitoring, making it an excellent option for developers looking to deploy web applications quickly and efficiently [18].

Storage

Storage is a critical component of any website hosting strategy, and AWS offers several options to cater to diverse needs. **S3** provides scalable object storage with high durability and availability, ideal for hosting static website content, media files, and backups [23]. **EFS** delivers scalable file storage for use with EC2 instances, facilitating the sharing of file data across multiple servers [23]. For long-term data archiving, **Glacier** offers a cost-effective storage solution with secure, durable, and low-cost storage designed for data that is infrequently accessed [23].

Database Services

Data management is vital for dynamic websites, and AWS offers a range of database solutions. **RDS** simplifies the setup, operation, and scaling of relational databases such as MySQL, PostgreSQL, Oracle, and SQL Server [21]. **DynamoDB** offers a fully managed NoSQL database service that

delivers single-digit millisecond performance at any scale and is a popular choice for applications requiring low latency [21]. **Aurora** is AWS's high-performance relational database service compatible with MySQL and PostgreSQL that delivers the performance and availability of high-end commercial databases at a fraction of the cost [21].

Networking Services

Efficient networking is crucial for the seamless delivery of web content. **VPC** enables users to reserve a logically isolated section of the AWS Cloud that provides enhanced security and network control [22]. **Route 53** is a scalable Domain Name System (DNS) service designed to route end-user requests to infrastructure running in AWS, guaranteeing high availability and low latency [22]. The **ELB** automatically distributes incoming application traffic across multiple targets, such as EC2 instances to ensure high fault tolerance and scalable performance [22].

Security & Compliance

AWS places significant emphasis on security and compliance, which are critical in today's digital landscape. **IAM** enables administrators to manage user access and permissions securely [24]. **Shield** provides protection against Distributed Denial of Service (DDoS) attacks, maintaining the availability of hosted websites [24]. **WAF** mitigates web applications from common web exploits [24]. Meanwhile, **CloudTrail** enables continuous monitoring and logging of account activity, contributing to enhanced security governance and compliance [24].

Deployment & DevOps

AWS supports modern DevOps practices through services that facilitate continuous integration and continuous delivery. **CodePipeline** automates the build, test, and deploy phases of application release to guarantee faster and more reliable updates [20]. **CodeDeploy** streamlines application deployments across a variety of compute services, reducing the risk of downtime during updates [20]. **CloudFormation** allows developers to model and provision AWS resources using templates to make infrastructure management both predictable and efficient [20].

3.2 List of AWS services used in website deployment

1. **VPC:** This service is used to isolate subnets, control routing, and keep internal traffic off the public Internet.
2. **EC2:** This service is used to host the Linux web servers that run Moodle, an admin "jump" instance for SSH access, and an application helper node.
3. **RDS:** This service is used to manage the data from Moodle in structured data without self-maintaining MySQL on EC2.
4. **EFS:** This service is used to provide a shared folder that every web server mounts so that any file uploaded on one instance is immediately visible to all other instances.
5. **Auto Scaling Group:** This service is used to monitor instance health and launch or terminate Web Server EC2 instances based on certain situations, ensuring flexible scaling during peak and low usage.
6. **ELB:** This service is used to route HTTP traffic across the Web Server instances and automatically remove any unhealthy node.
7. **AMI:** This service is used to copy an existing instance to create a new identical instance to be launched quickly for scaling or recovery.
8. **Security groups:** This service acts like a virtual firewall that allows only the required ports and sources to reach each instance or backend service.
9. **NAT gateway:** This service is used to allow outbound Internet access to private subnets while keeping them unreachable from outside.

3.3 Step by step website deployment

Step 1: Creating VPC

- **VPC Name:** ULearnlyVPC
- **CIDR Block:** 192.168.0.0/20
  - This provides 4,096 IP addresses.
- **IPv6 CIDR Block:** No IPv6 CIDR block
  - For this configuration, only IPv4 will be used.
- **Tenancy:** Default
  - This means the VPC is created on shared hardware.
- **Number of Availability Zones (AZs):** 2
  - 2 AZs are selected, which are the us-east-1a and us-east-1b.
- **Number of public subnets:** 2
  - These are used in each AZ to ensure that our web servers and load balancers are highly available. If one AZ experiences an issue, such as server failure, the traffic can be automatically routed to the other AZ to minimize the downtime.
- **Number of private subnets:** 2
  - These are used to ensure that our backend resources are distributed across AZs, which provides redundancy and fault tolerance. This ensures that the website can continue to function if one AZ encounters a failure.
- **NAT gateway:** In 1 AZ
  - A single NAT gateway in one AZ allows the private subnets in the VPC to access the Internet while minimizing costs. It allows outbound internet access for resources in the private subnets.
- **VPC endpoints:** None
  - No VPC endpoints are configured, meaning there is no direct access to S3 or other services within the VPC, and traffic will route through the internet if needed.
- **DNS options:** Enable DNS hostnames and DNS resolution
  - These 2 are enabled to allow internal DNS name resolution and external domain access within the VPC.

Details of VPC in AWS after the Configurations:

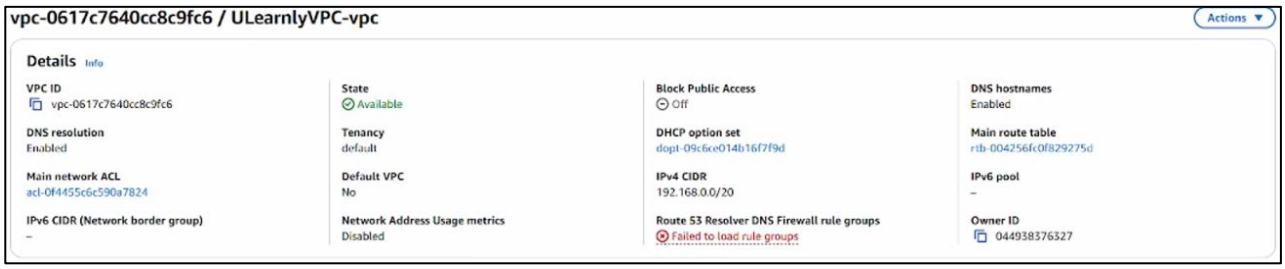


Figure 3.3.1 Details of VPC after configuration

VPC Architecture Diagram:

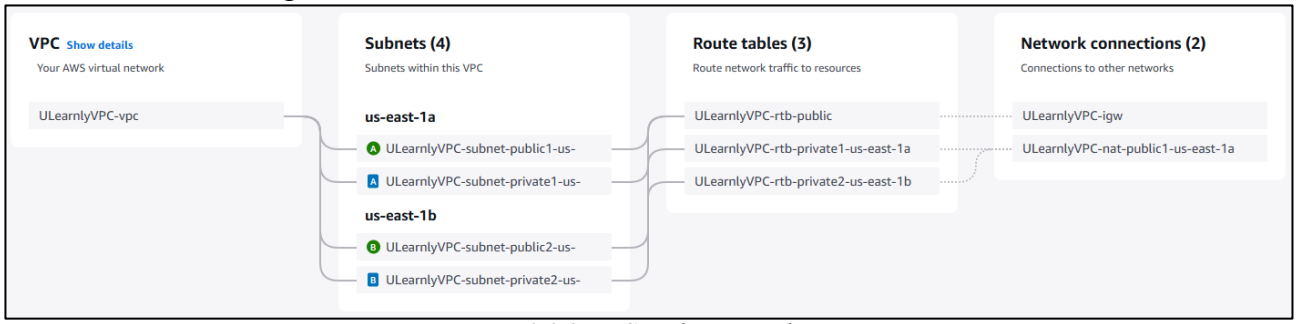


Figure 3.3.2 VPC architecture diagram

Step 2: Creating Security Groups

1) Creating a security group for the Web Server
   - **Security group name:** ULearnlyWebSG
   - **Description:** ULearnlyWebSG
   - **VPC:** Our custom VPC where resources for our online learning platform are hosted
   - **Inbound rules:**

Type	Protocol	Port Range	Source	CIDR Blocks
SSH	TCP	22	Anywhere-IPv4	0.0.0.0/0
HTTP	TCP	80	Anywhere-IPv4	0.0.0.0/0
HTTPS	TCP	443	Anywhere-IPv4	0.0.0.0/0

   - **Outbound rules:**

Type	Protocol	Port Range	Source	CIDR Blocks
All traffic	All	All	Custom	0.0.0.0/0

2) Creating a security group for the EFS
   - **Security group name:** ULearnlyEFSSG
   - **Description:** ULearnlyEFSSG
   - **VPC:** Our custom VPC where resources for our online learning platform are hosted
   - **Inbound rules:**

Type	Protocol	Port Range	Source	CIDR Blocks
NFS	TCP	2049	Custom	ULearnlyWebSG

   - **Outbound rules:**

Type	Protocol	Port Range	Source	CIDR Blocks
All traffic	All	All	Custom	0.0.0.0/0

3) Creating a security group for the App Server
   - **Security group name:** ULearnlyAppSG
   - **Description:** ULearnlyAppSG
   - **VPC:** Our custom VPC where resources for our online learning platform are hosted
   - **Inbound rules:**

Type	Protocol	Port Range	Source	CIDR Blocks
SSH	TCP	22	Anywhere-IPv4	0.0.0.0/0

   - **Outbound rules:**

Type	Protocol	Port Range	Source	CIDR Blocks
All traffic	All	All	Custom	0.0.0.0/0

4) Creating a security group for the Admin
   - **Security group name:** ULearnlyAdminSG
   - **Description:** ULearnlyAdminSG
   - **VPC:** Our custom VPC where resources for our online learning platform are hosted
   - **Inbound rules:**

Type	Protocol	Port Range	Source	CIDR Blocks
SSH	TCP	22	Anywhere-IPv4	0.0.0.0/0

   - **Outbound rules:**

Type	Protocol	Port Range	Source	CIDR Blocks
All traffic	All	All	Custom	0.0.0.0/0

5) Creating a security group for the Database
   - **Security group name:** ULearnlyDatabaseSG
   - **Description:** ULearnlyDatabaseSG
   - **VPC:** Our custom VPC where resources for our online learning platform are hosted
   - **Inbound rules:**

Type	Protocol	Port Range	Source	CIDR Blocks
MySQL/Aurora	TCP	3306	Custom	ULearnlyAppSG
MySQL/Aurora	TCP	3306	Custom	ULearnlyWebSG
MySQL/Aurora	TCP	3306	Custom	ULearnlyAdminSG

   - **Outbound rules:**

Type	Protocol	Port Range	Source	CIDR Blocks
All traffic	All	All	Custom	0.0.0.0/0

6) Configuring inbound rules for database access
   - **Inbound rules for Web Server:**

Type	Protocol	Port Range	Source	CIDR Blocks
MySQL/Aurora	TCP	3306	Custom	ULearnlyDatabaseSG

   - **Inbound rules for EFS:**

Type	Protocol	Port Range	Source	CIDR Blocks
MySQL/Aurora	TCP	3306	Custom	ULearnlyDatabaseSG

   - **Inbound rules for Admin:**

Type	Protocol	Port Range	Source	CIDR Blocks
MySQL/Aurora	TCP	3306	Custom	ULearnlyDatabaseSG

Step 3: Creating EFS

1) Creating file system
   - **Name:** ULearnlyEFS
   - **Virtual Private Cloud (VPC):** Our custom VPC where resources for our online learning platform are hosted

2) Customizing file system
   - **File system type:** Regional
     - Regional is set to store data across multiple Availability Zones (AZs) in the selected AWS Region. This also offers high availability and durability, which is ideal for applications and services requiring reliable file access like an online learning platform.
   - **Automatic backup:** Disabled
     - The file system won't be backed up by AWS Backup.
   - **Lifecycle management:**

	Transition into Infrequent Access (IA)	**Transition into Archive**	**Transition into Standard**
Option	None	None	None
Reason	All files in the file system may be accessed regularly and no file usage patterns are known so there's no benefit in moving them to IA storage.	Archived storage is suited for long-term retention of rarely accessed files. Since the system is still in the initial stage and actively used, there are no files considered "cold" enough to archive.	This option can only be activated when the other two options are enabled. Since no such transitions are enabled, this setting remains disabled by default.

   - **Encryption:** Enabled
     - This setting ensures that all files stored in the file system are encrypted using AWS Key Management Service (KMS) to protect our data.

3) Configuring performance settings
   - **Throughput mode:** Enhanced
     - This setting offers higher and more flexible throughput levels, making it appropriate for workloads with varying traffic and performance needs. It also ensures that the system can handle both high and low I/O automatically without human intervention.
   - **Elastic throughput:** Enabled
     - This setting automatically scales throughput based on activity related to workload. It also helps reduce costs by charging only for what is used, rather than a fixed provisioned rate.

4) Configuring network access
   - **Virtual Private Cloud (VPC):** Our custom VPC where resources for our online learning platform are hosted
   - **Mount targets:**

Availability Zone	Subnet ID	IP Address	Security Group
us-east-1a	ULearnlyVPC-subnet-public1-us-east-1a	Automatic	ULearnlyEFSSG
us-east-1b	ULearnlyVPC-subnet-public2-us-east-1b	Automatic	ULearnlyEFSSG

5) Configuring system policy
   - **Policy options:** Not configured (All left unchecked)
     - No custom system policy was defined. Default access behavior is used, custom permissions will be managed later if required.

Step 4: Creating Database

1) Choose a database creating method
   - **Database creating method:** Standard create
     - This setting allows full customization of database settings such as instance size, storage, security, and backup options. It provides more control compared to Easy Create.

2) Choosing engine options
   - **Engine type:** Aurora (MySQL Compatible)
   - **Engine version:** Aurora MySQL 3.05.2 (compatible with MySQL 8.0.32) – default for major version 8.0
   - **RDS Extended Support:** Disabled
     - This support is disabled since the selected engine version is still within the standard support period. Disabling it can avoid additional costs.

3) Choosing templates
   - **Template:** Production
     - This setting offers high availability, fast, and consistent performance and it is suitable for a live online learning platform.

4) Configuring database identifier and master credentials
   - **DB cluster identifier:** ULearnlyDatabase
   - **Master username:** admin
   - **Credentials management:** Self-managed
     - This setting allows more control over the login credentials, instead of using AWS Secrets Manager.
   - **Auto-generate password:** Unchecked
   - **Master password & Confirm master password:** admin123
     - A custom password is entered for this.

5) Configuring cluster storage
   - **Configuration options:** Aurora Standard
     - This setting offers cost-effective pricing, which is more suitable for applications with moderate I/O usage (where I/O costs less than 25% of total database costs).
     - This setting charges on a pay-per-request I/O basis, making it the best choice during the initial or moderate-use stage of the system.

6) Configuring instance
   - **DB instance class:** Burstable classes (includes t classes)
     - This class offers cost-effective performance for workloads that don't require consistently high CPU and are ideal for unpredictable traffic, which especially fits the needs of an online learning platform in its initial or growing stage.
   - **Selected type:** db.t3.medium
     - This instance offers 2 vCPUs and 4 GiB RAM with burstable performance. It is suitable for small to medium workloads with variable usage patterns.

7) Configuring availability & durability
   - **Multi-AZ deployment:** Don't create an Aurora Replica
     - This setting is to reduce costs and make a smart decision during the initial stage in a non-critical environment. However, it is recommended to enable it later for high availability and failover protection.

8) Configuring connectivity
   - **Compute resource:** Don't connect to an EC2 compute resource
     - No compute resources are attached. However, a manual connection will be set up later if required.
   - **Virtual private cloud (VPC):** Our custom VPC where resources for our online learning platform are hosted
   - **DB subnet group:** Create a new DB Subnet Group
     - This setting enables the DB cluster to access multiple subnets across different AZs for better fault tolerance.
   - **Public access:** No
     - This setting makes RDS not assign the public address to the cluster. Only allow EC2 instances and other resources within the VPC accessible to the cluster.
   - **VPC security group (firewall):** Choose existing VPC security groups
   - **Existing VPC security groups:** ULearnlyDatabaseSG
   - **Availability Zone:** No preference
   - **RDS Proxy:** Unchecked
     - This setting is skipped to reduce complexity and cost for now. However, it can be enabled later for better scalability, resiliency, and security.
   - **Certificate authority:** Default
     - This setting ensures secure, encrypted connections between applications and the database.
   - **Database port (under Additional configuration):** 3306
     - The default TCP/IP port is used by MySQL-compatible databases.

9) Configuring read replica write forwarding
   - **Local write forwarding:** Unchecked
     - This setting disallows write operations from read replicas, making the setup simple.

10) Configuring tags
   - **Tags:** Not added

11) Configuring database authentication
   - **IAM database authentication:** Unchecked
   - **Kerberos authentication:** Unchecked

12) Configuring monitoring
   - **Monitoring tools:** Database Insights – Standard
     - This database monitoring tool provides a basic view of database performance metrics for troubleshooting and analysis.
   - **Enhanced Monitoring (under Additional monitoring settings):** Unchecked
     - This setting skips monitoring for how processes and threads use the CPU.
   - **Log exports:** All left unchecked
     - This setting makes no logs are exported to the CloudWatch.
   - **IAM role:** Default

Step 5: Creating EC2

1) Creating EC2 for Web Server
   - **Name:** ULearnlyWebServer
   - **Amazon Machine Image (AMI):** Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
   - **Architecture:** 64-bit (x86)
     - This is the standard choice for general-purpose applications.
   - **Instance type:** t2.micro
     - This instance is Free Tier eligible and provides 1 vCPU and 1 GiB RAM, making it an ideal choice for basic workloads or initial deployment during development.
   - **Creating key pair:**
     - **Key pair name:** ULearnlyWebKey
     - **Key pair type:** RSA
       - This pair type is widely used that support and is compatible with most SSH clients and systems.
     - **Private key file format:** .pem
       - This format is used with OpenSSH commonly used on Unix/Linux-based systems such as Ubuntu.
   - **Configuring network settings:**
     - **VPC:** Our custom VPC where resources for our online learning platform are hosted
     - **Subnet:** ULearnlyVPC-subnet-public1-us-east-1a
     - **Auto-assign public IP:** Enabled
       - This setting allows the distribution of public IP address to instances automatically upon launch.
     - **Firewall (security group):** Existing security group
     - **Common security groups:** ULearnlyWebSG
   - **Configuring storage:**
     - **Volume size:** 8 GiB
     - **Volume type:** gp3 (General Purpose SSD)

AWS Diagram:

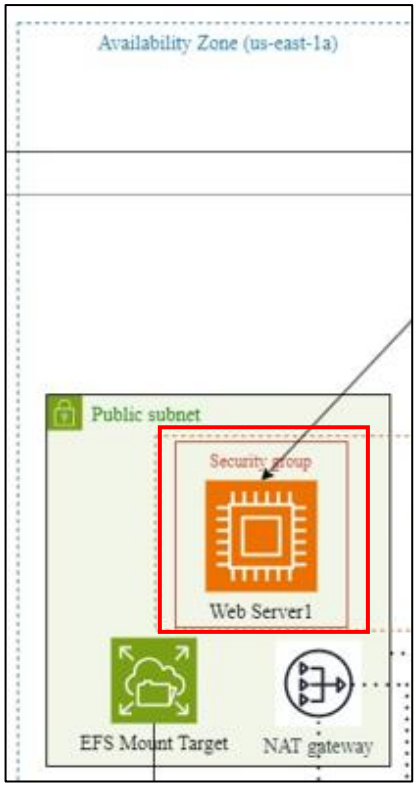


Figure 3.3.3 AWS Diagram

2) Creating EC2 for Admin
   - **Name:** ULearnlyAdmin
   - **Amazon Machine Image (AMI):** Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
   - **Architecture:** 64-bit (x86)
     - This is the standard choice for general-purpose applications.
   - **Instance type:** t2.micro
     - This instance is Free Tier eligible and provides 1 vCPU and 1 GiB RAM, making it an ideal choice for basic workloads or initial deployment during development.
   - **Creating key pair:**
     - **Key pair name:** ULearnlyAdminKey
     - **Key pair type:** RSA
       - This pair type is widely used that support and is compatible with most SSH clients and systems.
     - **Private key file format:** .pem
       - This format is used with OpenSSH commonly used on Unix/Linux-based systems such as Ubuntu.
   - **Configuring network settings:**
     - **VPC:** Our custom VPC where resources for our online learning platform are hosted
     - **Subnet:** ULearnlyVPC-subnet-public2-us-east-1b
     - **Auto-assign public IP:** Enabled
       - This setting allows the distribution of public IP address to instances automatically upon launch.
     - **Firewall (security group):** Existing security group
     - **Common security groups:** ULearnlyAdminSG
   - **Configuring storage:**
     - **Volume size:** 8 GiB
     - **Volume type:** gp3 (General Purpose SSD)

AWS Diagram:

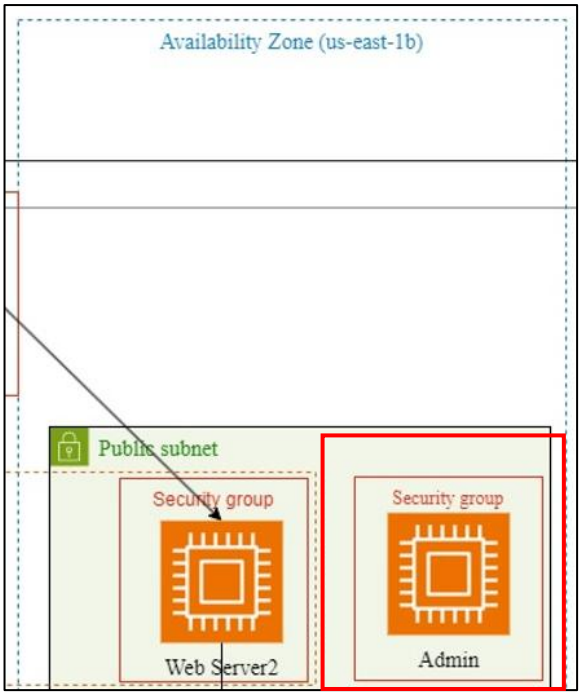


Figure 3.3.4 AWS Diagram

3) Creating EC2 for App
   - **Name:** ULearnlyApp
   - **Amazon Machine Image (AMI):** Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
   - **Architecture:** 64-bit (x86)
     - This is the standard choice for general-purpose applications.
   - **Instance type:** t2.micro
     - This instance is Free Tier eligible and provides 1 vCPU and 1 GiB RAM, making it an ideal choice for basic workloads or initial deployment during development.
   - **Creating key pair:**
     - **Key pair name:** ULearnlyAppKey
     - **Key pair type:** RSA
       - This pair type is widely used that support and is compatible with most SSH clients and systems.
     - **Private key file format:** .pem
       - This format is used with OpenSSH commonly used on Unix/Linux-based systems such as Ubuntu.
   - **Configuring network settings:**
     - **VPC:** Our custom VPC where resources for our online learning platform are hosted
     - **Subnet:** ULearnlyVPC-subnet-private1-us-east-1a
     - **Auto-assign public IP:** Disabled
       - This setting disallows the distribution of public IP address to instances automatically upon launch.
     - **Firewall (security group):** Existing security group
     - **Common security groups:** ULearnlyAppSG
   - **Configuring storage:**
     - **Volume size:** 8 GiB
     - **Volume type:** gp3 (General Purpose SSD)

AWS Diagram:

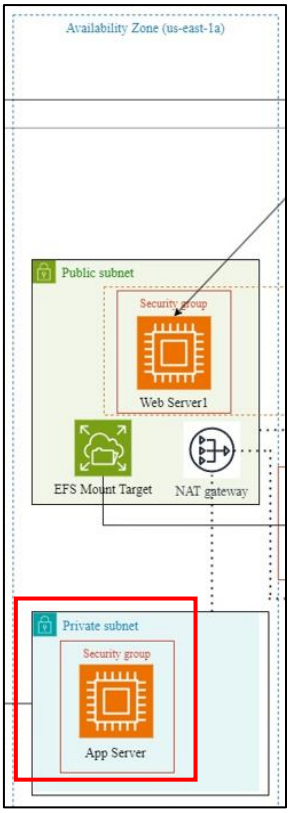


Figure 3.3.5 AWS Diagram

Step 6: Testing EC2 Instance Connectivity via SSH Using Key Pair Files

1) Organizing key files
   - The 3 key files (.pem) generated from instance creation were placed into a folder for better management and easier access later.
   - The 3 key files (.pem) are:
     1. ULearnlyAppKey.pem
     2. ULearnlyAdminKey.pem
     3. ULearnlyWebKey.pem
2) Opening terminal
   - A terminal window was opened and the working directory was changed to the directory containing the 3 key files (.pem).
3) Retrieving SSH connection command
   - The SSH connection command was copied from the "Connect" section under the SSH client tab of each EC2 instance in the AWS Management Console.
4) Establishing SSH connection
   - The copied SSH command (e.g., `ssh -i "ULearnlyWebKey.pem" ubuntu@public-dns`) was pasted and executed in the terminal to initiate a connection to the instance.
5) Verifying successful login
   - After running the SSH command, the terminal was successfully connected to the EC2 instance. As shown in Figure 3.3.6, the Ubuntu system displayed some system information.
   - Plus, the prompt changed to `ubuntu@ip-[ip_address]:~$`, confirming that the connection to the remote server was established successfully without errors.

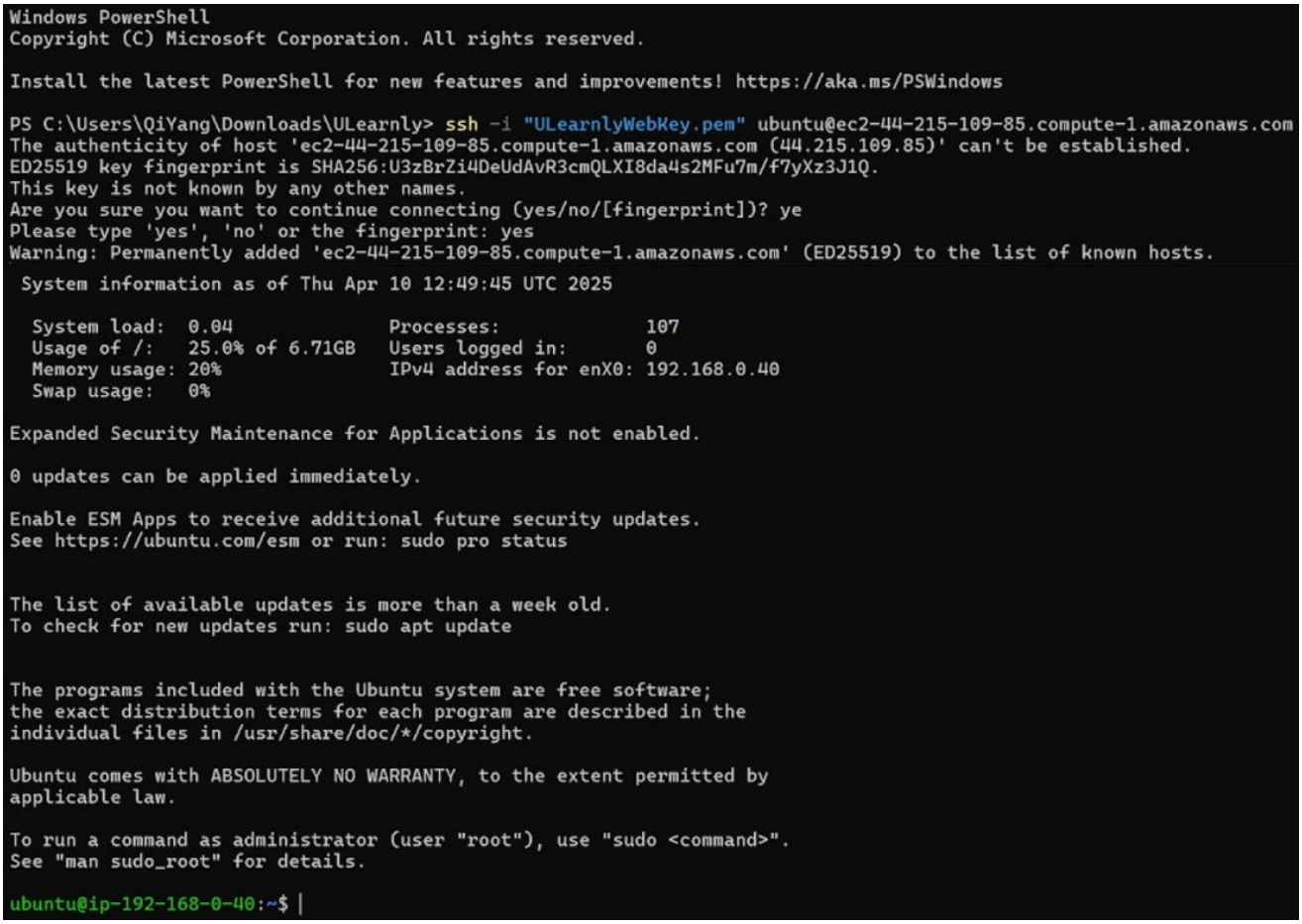


Figure 3.3.6 Command screenshot

Step 7: Connecting EC2 to EFS

1) Updating system packages and installing necessary software
   - **Commands used:**

No	Commands	Descriptions
1.	`sudo apt update`	Update the package list to ensure the latest versions of software are available for installation.
2.	`sudo apt install -y git`	Install Git, which is used to clone repositories such as the EFS utilities repository from GitHub.
3.	`git clone https://github.com/aws/efs-utils.git`	Clone the official EFS utilities repository needed to install the EFS mount helper for mounting the EFS file system.
4.	`sudo apt-get install -y gcc make python3-dev`	Install the necessary build tools (GCC, Make, Python3 development headers) needed to compile the EFS utilities from the source code.
5.	`sudo apt install -y pkg-config libssl-dev`	Install libraries (pkg-config, libssl-dev) needed for building secure SSL connections, which EFS utilities require.
6.	`curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh . "$HOME/.cargo/env"`	Download and install the Rust programming environment securely, then load the environment settings into the current terminal session.

2) Navigating to the EFS utils directory
   - **Command used:** `cd efs-utils`
     - This command is used to move into the efs-utils directory that contains the important source code and installation scripts.

3) Building and installing the EFS utils
   - **Command used:** `./build-deb.sh`
     - This command is used to install the EFS utilities for connecting the EC2 instance to EFS.

4) Installing the EFS utilities package
   - **Command used:** sudo apt-get -y install ./build/amazon-efs-utils*deb
     - This command is used to install the EFS utilities package that was just built, allowing the EC2 instance to mount and interact with Amazon EFS.

5) Creating the EFS mount point folder
   - **Command used:** `cd /`
     - This command is used to move to the root directory of the server before creating the EFS mount point.
   - **Command used:** `sudo mkdir moodledata-mount`
     - This command is used to create a new directory named moodledata-mount that serves as the mount point for the EFS.

6) Retrieving the EFS mount command
   1. In the AWS Management Console, go to the Elastic File System service.
   2. Click on “File systems” in the left sidebar.
   3. Select the ULearnlyEFS’s File system ID.
   4. Click the "Attach" button on the top right.
   5. Copy the command shown under "Using the EFS mount helper".

7) Mounting the EFS file system
   - Before pasting, the default mount point /efs should be replaced with moodledata-mount/ to match the created folder.
   - **Example command:** `sudo mount -t efs -o tls fs-096667a4e99601a87:/ moodledata-mount`

8) Checking the connectivity (Optional Step)
   - **Command used:** `df -h`
     - This command is used to display all mounted file systems and their usage.
   - **Example output:**

```
ubuntu@ip-192-168-0-40:/$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       6.8G  4.0G  2.8G  60% /
tmpfs           479M     0  479M   0% /dev/shm
tmpfs           192M  964K  191M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/xvda16     881M   79M  741M  10% /boot
/dev/xvda15     105M  6.1M   99M   6% /boot/efi
tmpfs            96M   12K   96M   1% /run/user/1000
127.0.0.1:/     8.0E     0  8.0E   0% /moodledata-mount
```

Figure 3.3.7 Command output

As shown in Figure 3.3.7, the output shows that the EC2 instances have successfully mounted the EFS to the moodledata-mount/ directory.

Step 8: Installing Moodle

1) Updating system packages and installing necessary software
   - **Command used:** `sudo apt install -y mysql-client mysql-server apache2 php php-mysql php-mbstring php-xml php-curl php-zip php-gd php-intl php-soap`
     - This command is used to install software packages for setting up a web server and database environment, including MySQL, Apache2, and so on to support applications such as Moodle.

2) Downloading and extracting Moodle
   - **Command used:** `cd /var/www/html`
     - This command is used to move to the default web root directory where Apache serves website files.
     - This is also where Moodle will be installed.
   - **Command used:** `sudo wget` https://download.moodle.org/download.php/direct/stable405/moodle-latest-405.tgz
     - This command is used to download the Moodle package.
     - The download link is copied from the official Moodle website.
   - **Command used:** `sudo tar -xvzf moodle-latest-405.tgz`
     - This command is used to unzip the downloaded file and unpack it into the /var/www/html directory, making the application ready for configuration and installation through the web browser.

3) Changing permissions to allow Moodle to access the folders
   - **Command used:**
     ```
     sudo chmod -R 777 moodle
     sudo chown www-data moodle
     ```
     - These commands are used to set full read, write, and execute permissions to all users on the moodle folder and change the ownership of the moodle folder to Apache web server user.
   - **Command used:**
     ```
     sudo chown -R 777 /moodledata-mount/
     sudo chmod www-data /moodledata-mount/
     ```
     - These commands are used to set full permissions on the /moodledata-mount/ directory and change the ownership of the directory to the Apache web server user so that the server can access and manage files stored in the EFS.

4) Accessing the Apache2 default web page via a web browser
   - Copy the Public IPv4 address from the ULearnlyWebServer EC2 instances' details page in the AWS Console.
   - Open a web browser and add "HTTP://" before pasting the Public IPv4 address.
   - After that, the Apache2 Ubuntu Default Page is displayed, indicating that the EC2 instance is successfully hosting the web server.

5) Accessing the Moodle setup page
   - Add "/moodle" after the Public IPv4 address in the web browser URL.
   - As shown in Figure 3.3.8, the Moodle setup page is successfully opened.

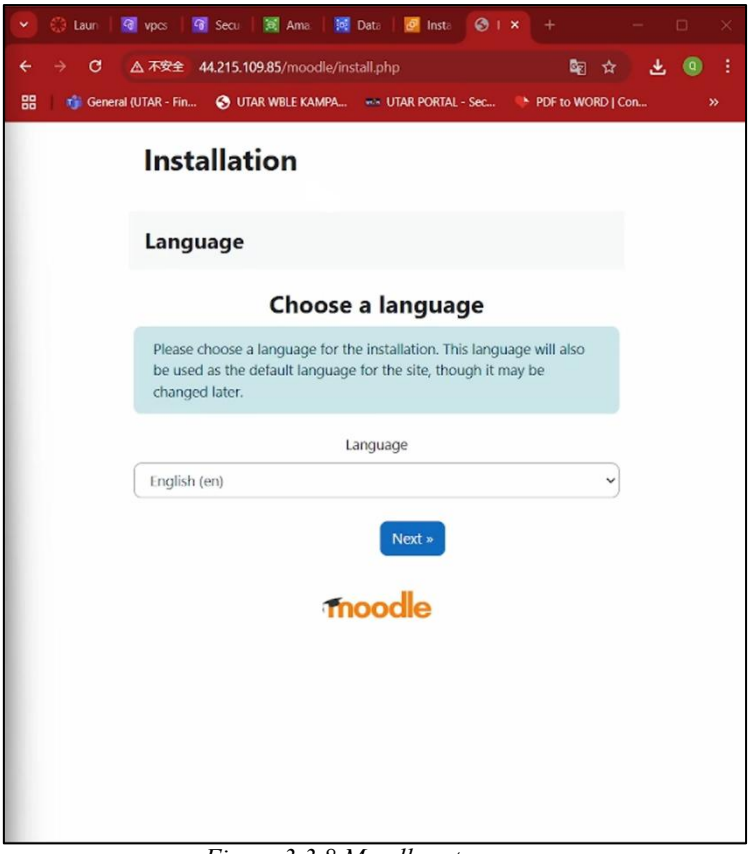


Figure 3.3.8 Moodle setup page

6) Configuring the Moodle installation settings
   - **Confirming paths:**
     - **Data directory:** /moodledata-mount/
       - This specifies that Moodle will store the data (such as course files, user uploads, and backups) in the /moodledata-mount/ directory, which is mounted from the Amazon EFS.
   - **Choosing database driver:**
     - **Database driver type**: Aurora MySQL (native/auroramysql)
       - Moodle can then connect directly to the RDS Aurora database.
   - **Configuring database settings:**
     - **Database host:** Copy the Endpoint from the Connectivity & Security section of the AWS RDS Aurora console
     - **Database name:** moodle
     - **Database user:** admin
     - **Database password:** admin123
     - **Tables prefix:** mdl_
     - **Database port:** 3306
     - **Unix socket:** Leave it blank
   - **Adjusting PHP Settings for Moodle requirements:**
     - **Command used:** `sudo nano /etc/php/8.3/apache2/php.ini`
       - This command is used to edit the PHP configuration file.
     - Search for max_input_vars and then remove the semicolon of that line and change the value to 5000.
   - **Restarting Apache:**
     - **Command used:** `sudo systemctl restart apache2`
       - This command is used to apply the new settings to the Apache.

- **Configuring main administrator account:**
  - **Username:** admin
  - **New password:** Admin123@
  - **First name:** Admin
  - **Last name:** User
  - **Email address:** personal@email.com
  - **Email visibility:** Visible to everyone
- **Configuring site home settings:**
  - **Full site name:** ULearnly Online Learning
  - **Short name for the site:** ULearnly
- **Configuring support contact:**
  - **Support email:** personal@email.com
- **Configuring registration information:**
  - **Country:** Malaysia
  - **Admin email address:** personal@email.com
  - **Privacy notice and data processing agreement:** Checked

After completing the Moodle installation:

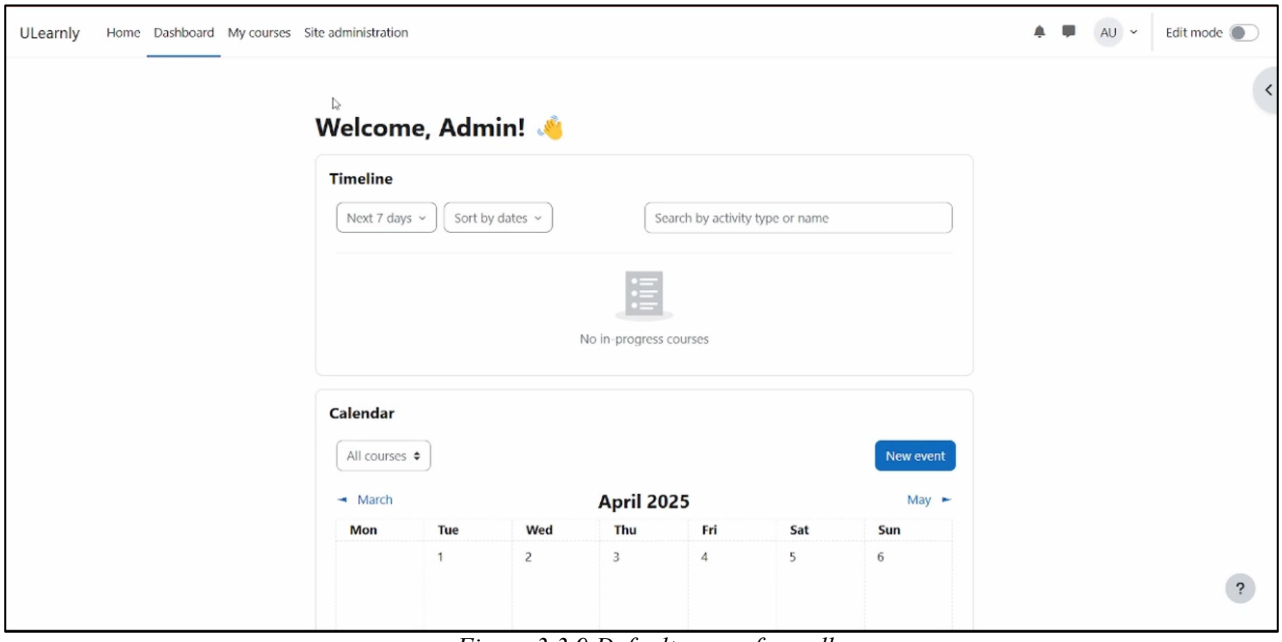


Figure 3.3.9 Default page of moodle

<u>Step 9: Using the Admin Server as a Jump Host to Access the Application Server</u>

1) Retrieving Admin Server's SSH connection command
   - The SSH connection command was copied from the "Connect" section under the SSH client tab of each EC2 instance in the AWS Management Console.

2) Establishing SSH connection to the Admin Server
   - A terminal window was opened and the working directory was changed to a directory containing the 3 key files (.pem).
   - The copied SSH command (e.g., `ssh -i "ULearnlyAdminKey.pem" ubuntu@public-dns`) was pasted and executed in the terminal to initiate a connection to the instance.

3) Uploading the Application Server key to the Admin Server
   - A new terminal window was opened.
   - **Command used:** `scp -i "path/to/ULearnlyAdminKey.pem" path/to/ULearnlyAppKey.pem ubuntu@<Admin-Server-Public-IP>:~`
     - This command is used to securely transfer the Application Server's private key to the home directory of the ubuntu user on the Admin Server.

4) Checking transferred files on the Admin Server (Optional Step)
   - **Command used:** `ls`
     - From the home directory, the ls command is used to check the existence of the file transferred.

5) Protecting the Application Server key
   - **Command used:** `chmod 400 ULearnlyAppKey.pem`
     - This command is used to protect the App Server key by making it readable only by the file owner.

6) Connecting to the Application Server from the Admin Server
   - Paste the SSH command (e.g., `ssh -i "ULearnlyAdminKey.pem" ubuntu@public-dns`) copied from the "Connect" section under the SSH client tab of Admin Server's EC2 instances in the AWS Management Console on the Admin Server's terminal.

7) Updating system packages and installing MariaDB Server
   - **Command used:** `sudo apt update`
     - This command is used to update the package lists.
   - **Command used:** `sudo apt install -y mariadb-server`
     - This command is used to install the MariaDB Server package, which will be used as the database server for Moodle.

8) Connecting to the RDS Database
   - **Command used:** `mysql -h ULearnlydatabase-instance-1.czhepcndkhpy.us-east-1.rds.amazonaws.com -P 3306 -u admin -p`
     - This command contains the RDS-endpoint copied from the Connectivity & Security section of the AWS RDS Aurora console and the database username.
   - **Password:** admin123

9) Verifying database connectivity to the RDS instance (Optional Step)
   - **Command used:** `SHOW DATABASES;`

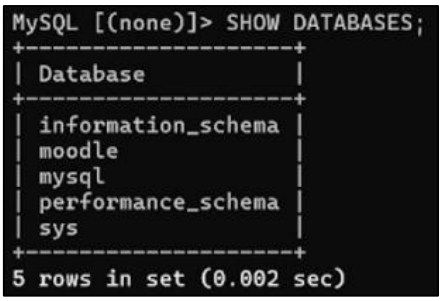


Figure 3.3.10 Command output

Step 10: Create an AMI Backup of the Web Server Instance

1) Creating an image from the Web Server's EC2 instance
   - In the AWS EC2 Console, select ULearnlyWebServer instance → Image and templates → Create image.
   - **Configuring image:**
     - **Image name:** ULearnlyWebServerImage
     - **Image description:** ULearnlyWebServerImage
     - **Reboot instance:** Checked
       - This setting is enabled to ensure that the file system is consistent and that all data is properly written to disk before the AMI is created.
     - **Instance volume size:** 8 GiB
     - **Delete on termination:** Enabled
       - This setting automatically deletes the volume when the associated EC2 instance is terminated.
     - **Tags:** Tag image and snapshots together
       - This setting shares the same tags with both the AMI and its associated snapshots.

2) Launching a new EC2 instance from the created AMI
   - In the AWS EC2 Console, select AMIs under the Images from the left sidebar.
   - Select the ULearnlyWebServerImage to create an instance.
   - **Configuring instance:**
     - **Name:** ULearnlyWebServer2
   - **Configuring key pair:**
     - **Key pair name:** ULearnlyWebKey
   - **Configuring network settings:**
     - **VPC:** Our custom VPC where resources for our online learning platform are hosted
     - **Subnet:** ULearnlyVPC-subnet-public2-us-east-1b
     - **Auto-assign public IP:** Enabled
       - This setting allows the distribution of public IP address to instances automatically upon launch.
     - **Firewall (security group):** Existing security group
     - **Common security groups:** ULearnlyWebSG
   - **Configuring storage:**
     - **Volume size:** 8 GiB
     - **Volume type:** gp3 (General Purpose SSD)

AWS Diagram:

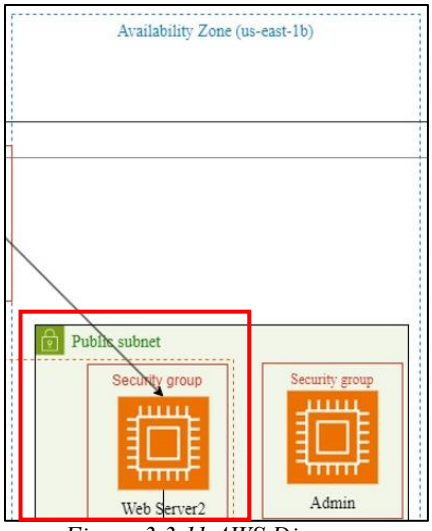


Figure 3.3.11 AWS Diagram

Step 11: Creating Load Balancer

1) Creating target groups
   - **Specifying group details:**
     - **Target type:** Instances
     - **Target group name:** ULearnlyWebInstances
     - **Protocol:** HTTP
     - **Port:** 80
     - **IP address type:** IPv4
     - **VPC:** Our custom VPC where resources for our online learning platform are hosted
     - **Protocol version:** HTTP1
   - **Configuring health checks:**
     - **Health check protocol:** HTTP
     - **Health check path:** /
   - **Registering Web Server instances to the Load Balancer target group:**
     - Select both ULearnlyWebServer and ULearnlyWebServer2 and register into the target group.

2) Creating load balancer
   - **Choosing load balancer type:**
     - **Load balancer type:** Application Load Balancer
   - **Configuring load balancer:**
     - **Load balancer name:** ULearnlyELB
     - **Scheme:** Internet-facing
     - **Load balancer IP address type:** IPv4
   - **Configuring network mapping:**
     - **VPC:** Our custom VPC where resources for our online learning platform are hosted
     - **IP pools:** Unchecked
     - **Availability zones and subnets:** Checked us-east-1a & us-east-1b
   - **Configuring security groups:**
     - **Security groups:** ULearnlyWebSG
   - Configuring listeners and routing:
     - **Protocol:** HTTP
     - **Port:** 80
     - **Default action:** ULearnlyWebInstances

Checking if EFS is Working

1) Opening 2 terminals: Web Server 1 and Web Server 2

Web Server 1:	**Web Server 2:**
• Navigating to /moodledata-mount/ – **Command used:** `cd /moodledata-mount/`	• Navigate to /moodledata-mount/ – **Command used:** `cd /moodledata-mount/`
• Creating a random file – **Command used:** `nano hello.txt`	
	• Checking if the file exists – **Command used:** `ls`

Example of verifying EFS connectivity between two servers:

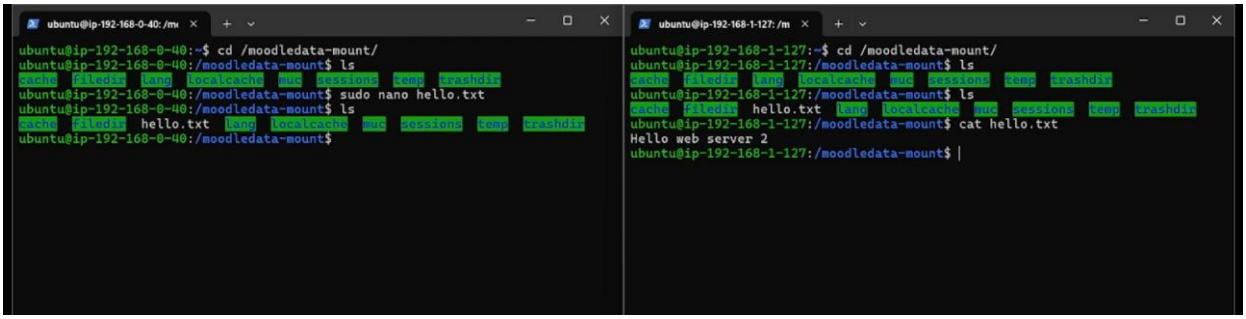


Figure 3.3.12 Command output

Checking if ELB is Working

Normal Healthy Situation

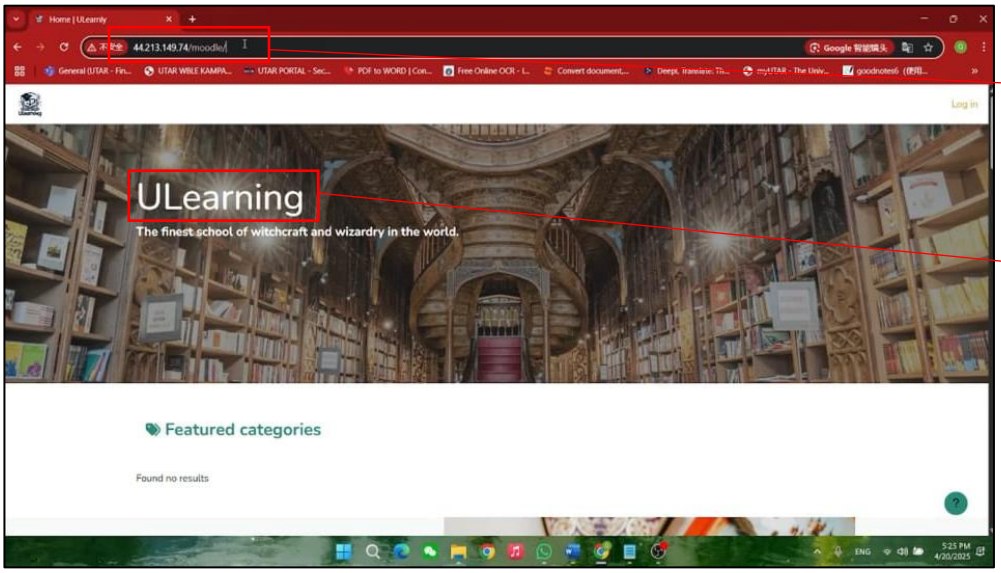
1) Copy the DNS name of the Load Balancer
2) Paste the copied DNS name into the web browser's address bar
3) Enter the one with /moodle at the end of the DNS name URL

Special Situation with One Instance Fails (Stops)

1) Stop ULearnlyWebServer
2) Repeat the same steps from the normal healthy situation

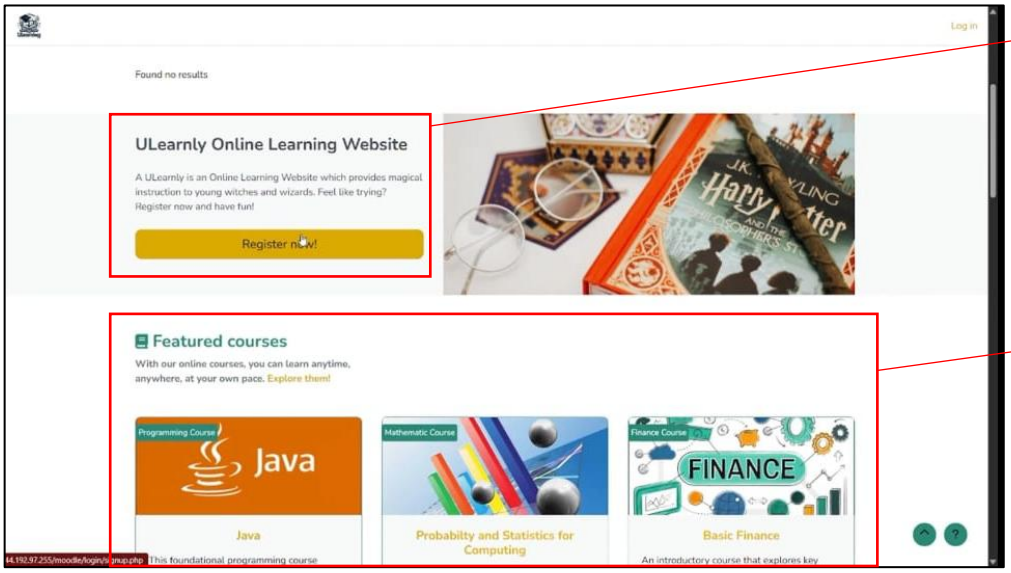
Conclusion: Even if one of the web server instances stops, the Moodle application remains accessible. This concludes that the Elastic Load Balancer (ELB) is functioning correctly by automatically routing traffic to the other instances, ensuring high availability and minimizing service disruption.

Final ULearnly Online Learning Website



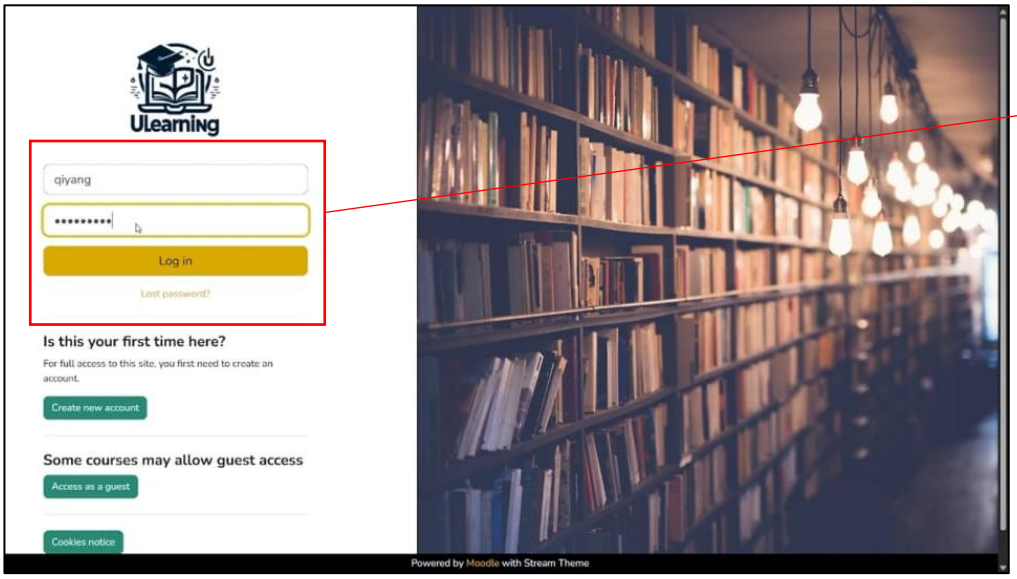
Link to ULearnly website

Proposed name for online learning website

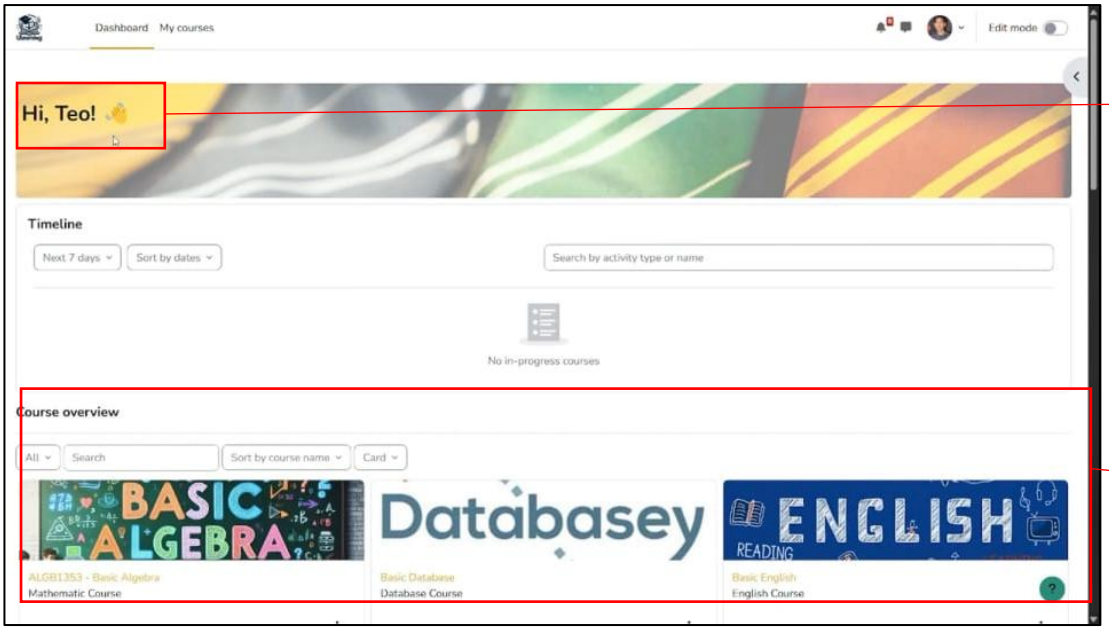


Description for online learning website

Available courses

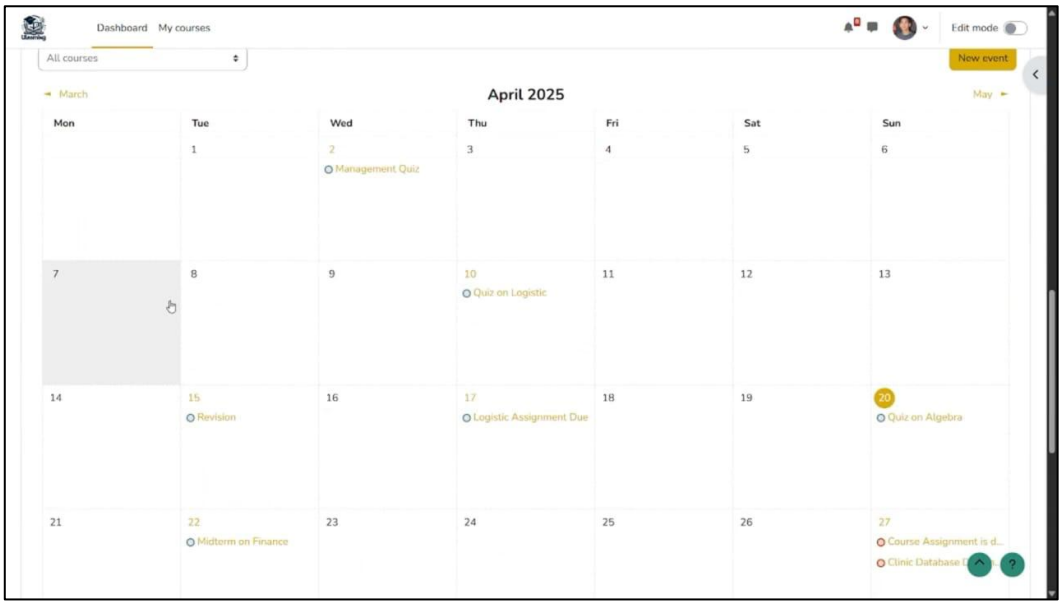


Login page

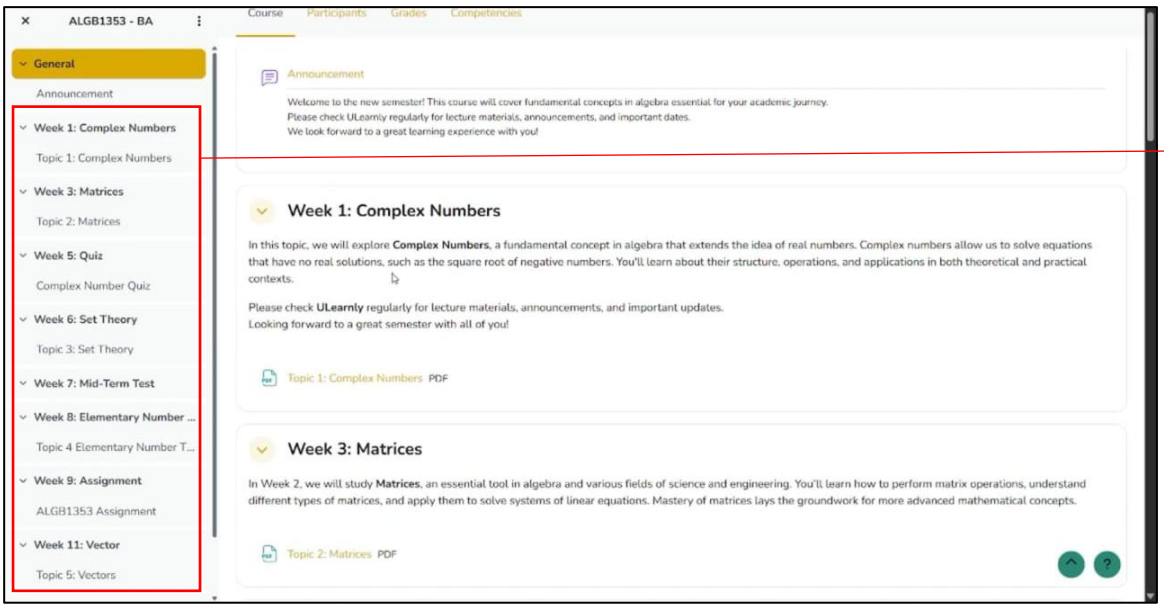


Student name

Course enrolled

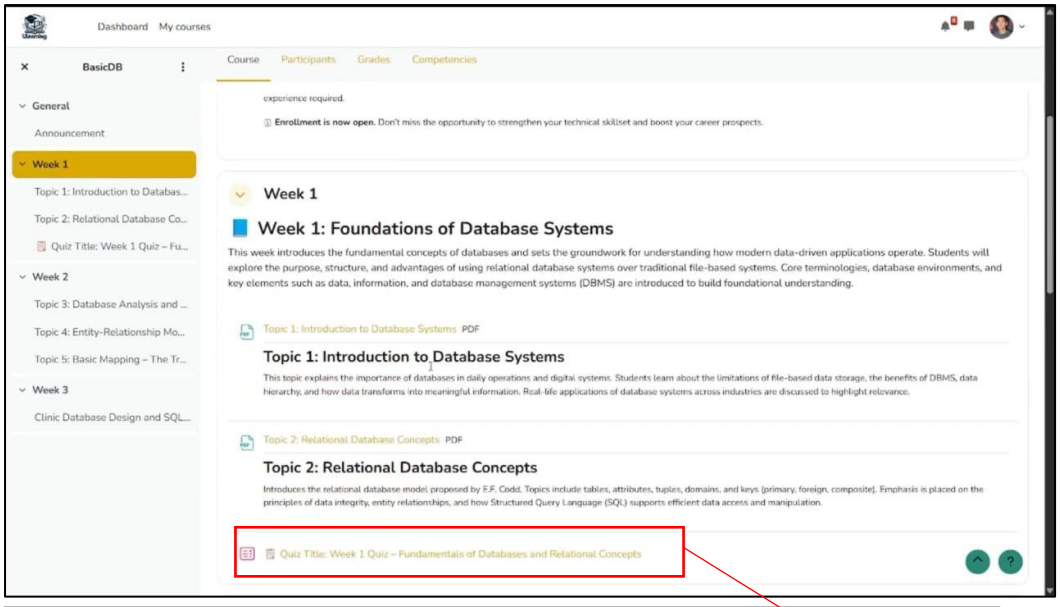


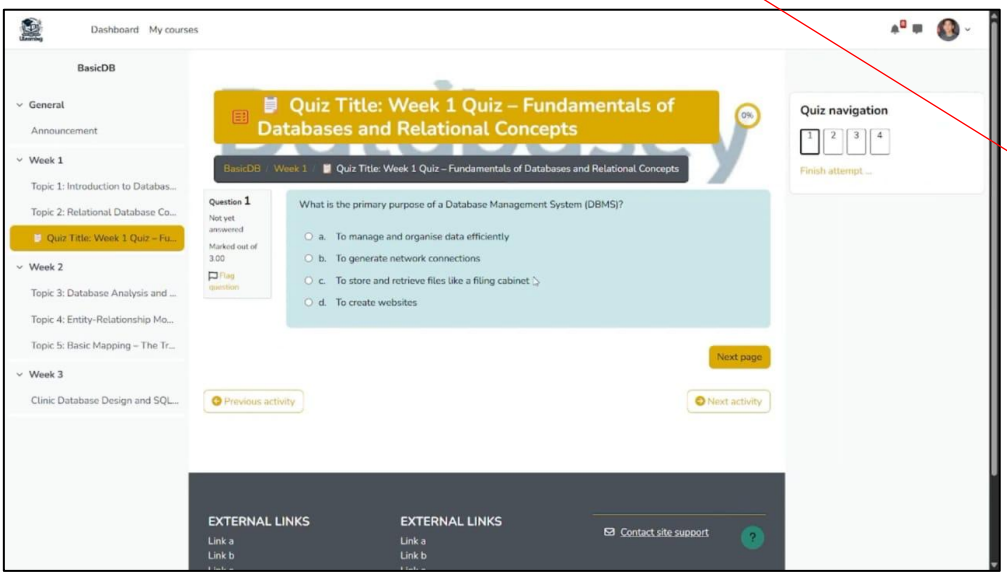
Personalised calendar



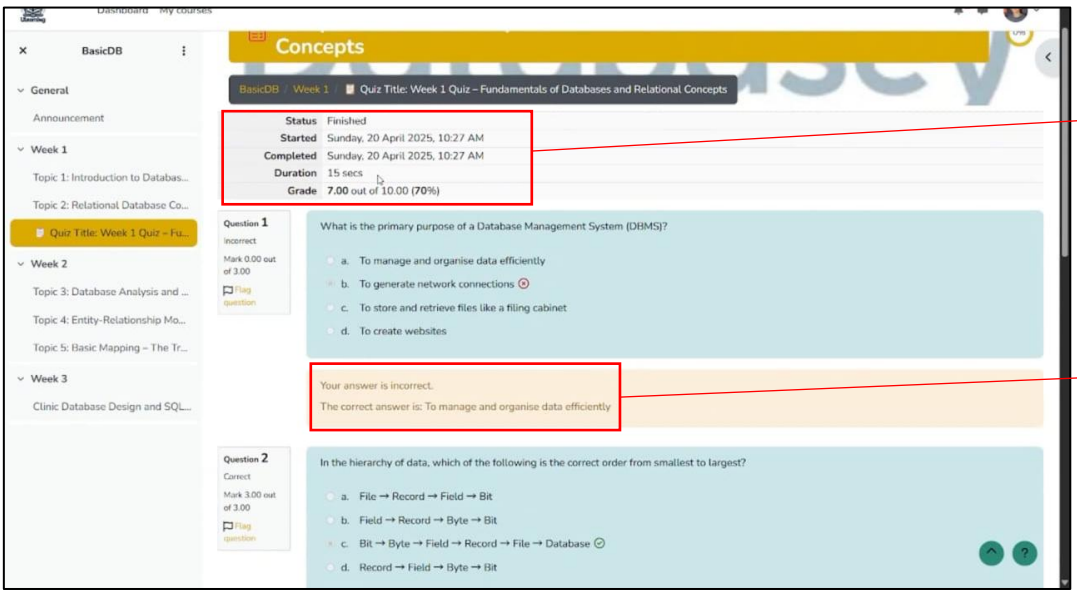
View course timeline

View course content



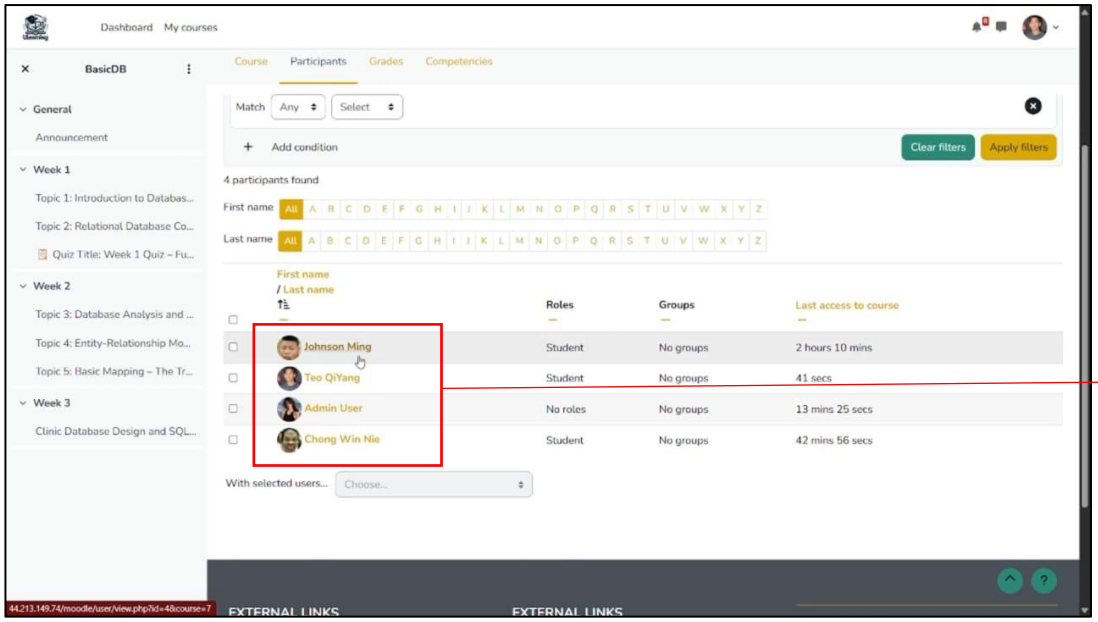


Attempt quiz



View grade after attempt

View correct answer



3.4 Solution architecture diagram

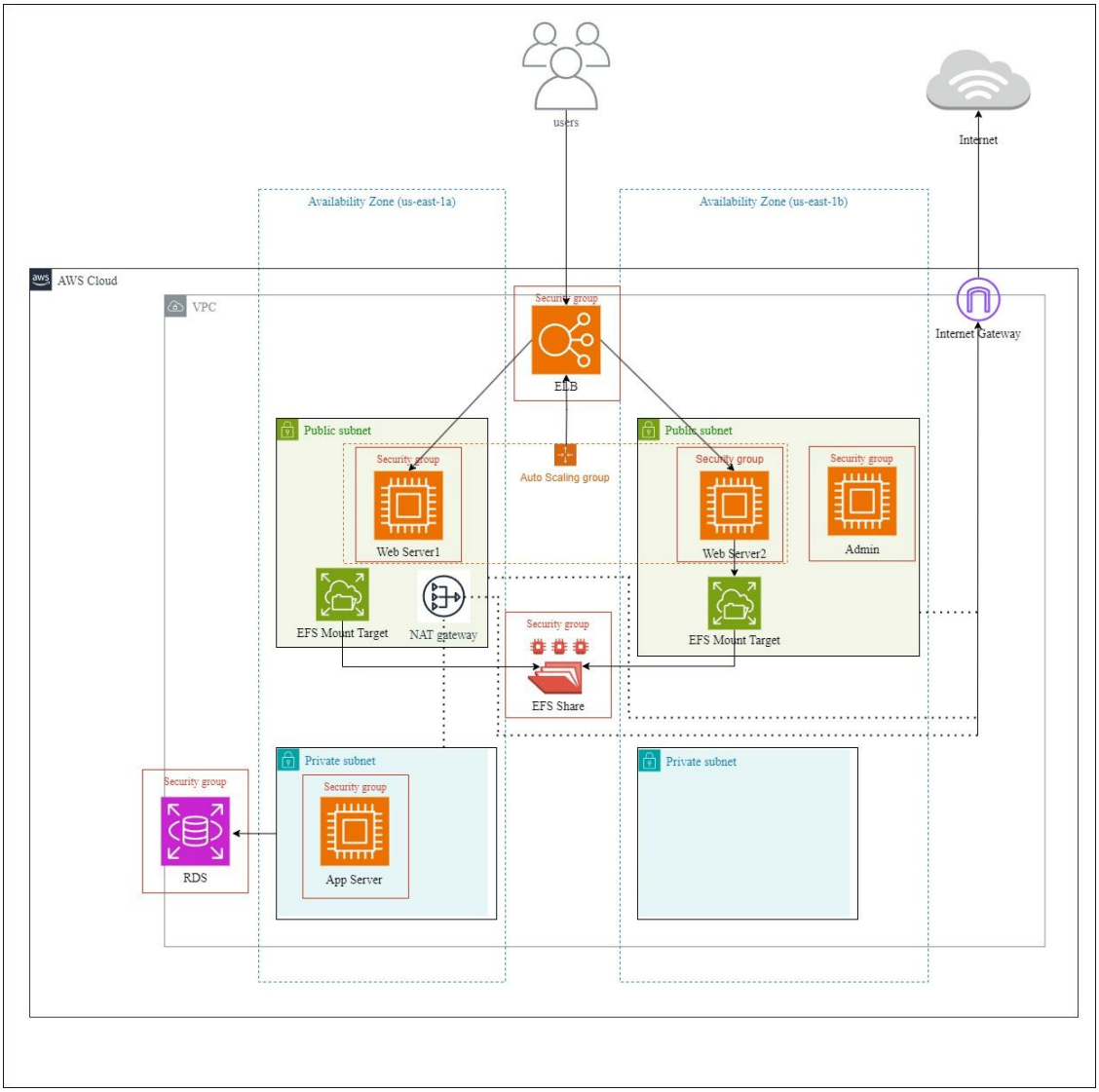


Figure 3.4.1 Solution architecture diagram

CHAPTER 4: CONCLUSION AND FUTURE WORKS

4.1 Conclusion

This report has examined the implementation of a cloud-based online learning platform hosted on AWS, highlighting its scalability, security, and cost-efficiency. Through a comparative analysis with other leading cloud providers, Google Cloud, Microsoft Azure, and IBM Cloud, AWS was found to offer the most comprehensive and flexible suite of services suitable for educational environments. Real-world case studies, such as those of Airbnb, Netflix, and NASA, further validate AWS's capability to support high-demand, globally accessible web solutions.

Advanced considerations such as data protection, compliance, pricing flexibility, and sustainability underscore AWS's maturity as a cloud infrastructure provider. The platform not only addresses current operational needs but also positions institutions for future growth through reliable, scalable, and environmentally conscious technologies. Therefore, AWS is presented as a highly suitable choice for deploying and managing modern e-learning platforms that demand availability, security, and performance. By addressing these gaps through strategic adoption of AWS services and continuous innovation, educators can unlock the full potential of cloud-based learning, ensuring equitable and future-ready education for all.

4.2 Future Works

The integration of AWS into e-learning platforms presents immense opportunities for innovation. Future research could explore leveraging **AWS AI/ML services like SageMaker** to create adaptive learning systems that personalize content based on student performance. Additionally, hybrid cloud architecture could bridge gaps for institutions transitioning from legacy systems. Cost-optimization frameworks using **AWS Trusted Advisor and reserved instances** warrant deeper analysis to maximize ROI (Return on Investment). Enhancing accessibility through AWS's edge computing and IoT integrations could further democratize education in underserved regions. Finally, advancing security protocols using **AWS Shield and automated compliance tools** will be critical as cyber threats evolve.

Possible features to be added in the future:

- Blockchain for secure credentialing: Blockchain technology will revolutionize academic credentialing, allowing students to store and share verifiable digital certificates securely. Cloud-based blockchain platforms will prevent fraud and simplify verification processes.
- Integration of 5G for Faster Access: With the rollout of 5G technology, cloud-based education will become even more efficient, enabling low-latency streaming, high-quality video lectures, and real-time interactions.
- Increased Use of Virtual and Augmented Reality: Cloud computing will facilitate the growth of VR and AR in education, providing more immersive learning experiences across various disciplines. For example, medical students will be able to perform virtual surgeries before handling real patients. Besides, history students can explore ancient civilizations through VR simulations.
- Easily integrates AI (chatbots, learning paths): AI-powered virtual tutors and chatbots will become more advanced, providing real-time assistance to students. These AI systems will answer questions, track progress, and recommend personalized study plans.

References

[1] Coursera, "Serving the world through learning impact report," 2021. Available: https://about.coursera.org/press/wp-content/uploads/2021/11/2021-Coursera-Impact-Report.pdf

[2] LMSPortals, "How Cloud Computing is Transforming eLearning: The Future of Digital Education," *lmsportals*, Feb. 09, 2025. https://www.lmsportals.com/post/how-cloud-computing-is-transforming-elearning-the-future-of-digital-education

[3] I. Limited, "The Benefits of Cloud Computing in Education | Infosys BPM | Infosys BPM," www.infosysbpm.com. https://www.infosysbpm.com/blogs/education-technology-services/cloud-computing-benefits-in-education.html

[4] AWS, "Overview of Amazon Web Services," *Amazon.com*, Aug. 27, 2024. https://docs.aws.amazon.com/whitepapers/latest/aws-overview/introduction.html

[5] AWS, "AWS Global Infrastructure," *Amazon Web Services, Inc.*, 2024. https://aws.amazon.com/about-aws/global-infrastructure/

[6] "NASA Case Study – Amazon Web Services (AWS)," *Amazon Web Services, Inc.*, 2011. https://aws.amazon.com/partners/success/nasa-image-library/

[7] AWS, "Netflix Case Study – Amazon Web Services (AWS)," Amazon Web Services, Inc., 2016. https://aws.amazon.com/solutions/case-studies/netflix-case-study/

[8] AWS, "Airbnb Case Study – Amazon Web Services (AWS)," *Amazon Web Services, Inc.*, 2019. https://aws.amazon.com/solutions/case-studies/airbnb-case-study/

[9] Amazon, "Benefits," Amazon Web Services, Inc. https://aws.amazon.com/application-hosting/benefits/

[10] Google, "Google Cloud overview | Overview," Google Cloud, 2023. https://cloud.google.com/docs/overview

[11] Microsoft, "What is Azure—Microsoft Cloud Services | Microsoft Azure," *azure.microsoft.com*, 2025. https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure

[12] IBM, "IBM Cloud," *Ibm.com*, 2019. https://www.ibm.com/cloud

[13] "E-Commerce with Shopify – Amazon Web Services," *Amazon Web Services, Inc.* https://aws.amazon.com/eventbridge/integrations/shopify/

[14] "SHOPLINE Case Study," *Amazon Web Services, Inc.*, 2021. https://aws.amazon.com/solutions/case-studies/shopline/

[15] Yadavpuneet, “Why Does Facebook Use Amazon Web Services (AWS) for its Infrastructure?,” Medium, Aug. 11, 2023. https://medium.com/@yadavpuneet2001/why-does-facebook-use-amazon-web-services-aws-for-its-infrastructure-dc9759d44134

[16] B. A. info @bb.agency and C. Slingerland, “How Much Does Twitter Spend On AWS And Google Cloud?,” *CloudZero*, Aug. 15, 2023. https://www.cloudzero.com/blog/twitter-aws/

[17] G. Times, “Does Reddit use aws? - GB Times,” GB Times, Sep. 07, 2024. https://gbtimes.com/does-reddit-use-aws/

[18] “Expedia Group launches travel portal in months using the AWS Cloud,” Amazon Web Services, Inc. Available: https://aws.amazon.com/products/compute/

[19] Vasile Crudu, “What are some common challenges faced by AWS Developers?,” Moldstud.com, Dec. 07, 2024. https://moldstud.com/articles/p-what-are-some-common-challenges-faced-by-aws-developers

[20] “DevOps - Amazon Web Services (AWS),” Amazon Web Services, Inc. https://aws.amazon.com/devops/

[21] “Choosing an AWS database service - Choosing an AWS database service,” docs.aws.amazon.com. https://docs.aws.amazon.com/decision-guides/latest/databases-on-aws-how-to-choose/databases-on-aws-how-to-choose.html

[22] AWS, “Networking and Content Delivery – Amazon Web Services,” Amazon Web Services, Inc., 2018. https://aws.amazon.com/products/networking/

[23] “AWS re:Invent 2023 Innovation Talk - AWS storage: The backbone for your data-driven business,” Amazon Web Services, Inc. Available: https://aws.amazon.com/products/storage/

[24] “Cloud Security, Identity, and Compliance Products – Amazon Web Services (AWS),” Amazon Web Services, Inc., 2025. https://aws.amazon.com/products/security/?nc=sn&loc=2&refid=b528af74-6453-48fc-b5e3-4d0ff57d1f23

[25] Amazon, “Compliance Programs - Amazon Web Services (AWS),” Amazon Web Services, Inc., 2010. https://aws.amazon.com/compliance/programs/

[26] AWS, “AWS Product and Service Pricing | Amazon Web Services,” Amazon Web Services, Inc., 2024. https://aws.amazon.com/pricing/?aws-products-pricing.sort-by=item.additionalFields.productNameLowercase&aws-products-pricing.sort-order=asc&awsf.Free%20Tier%20Type=

[27] AWS, “Technologies for Sustainability – Amazon Web Services,” Amazon Web Services, Inc. https://aws.amazon.com/sustainability/

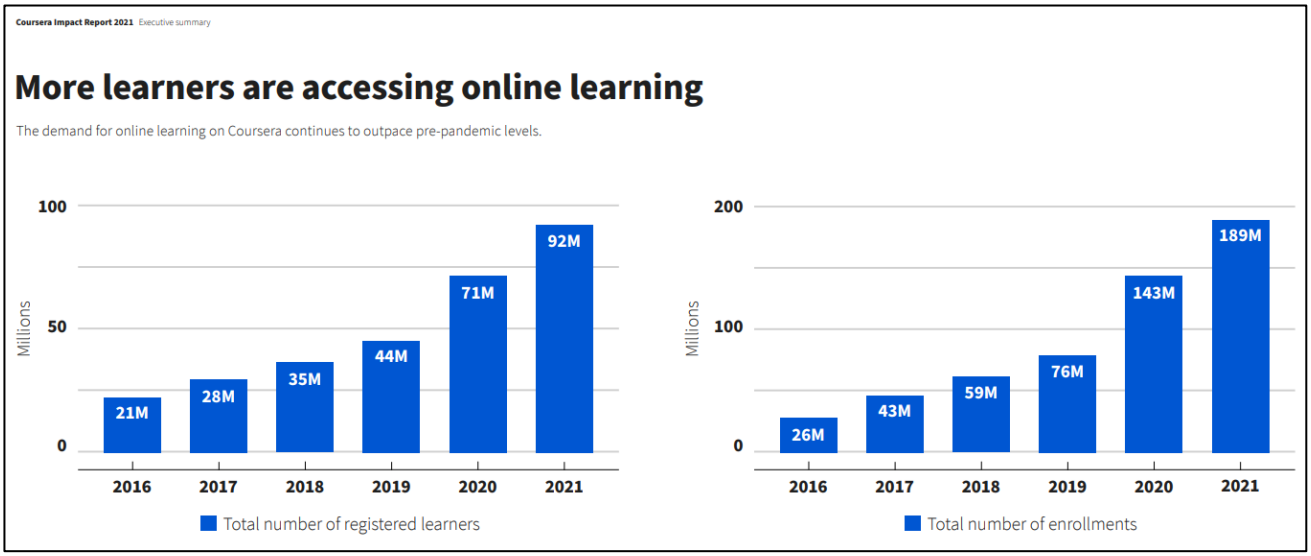
Appendixes

Appendix 1 - List of Abbreviation

Abbreviation	Meaning
AWS	Amazon Web Services
PDPA	Personal Data Protection Act 2010
GCP	Google Cloud Platform
IBM	International Business Machines
AZ	Availability zone
VM	Virtual machines
IAM	Identity access and management
ISO	International Organization for Standardization
SOC	System and Organization Controls
HIPAA	Health Insurance Portability and Accountability Act
GDPR	General Data Protection Regulation
FedRAMP	Federal Risk and Authorization Management Program
PCI DSS	Payment Card Industry Data Security Standard
SSH	Secure Socket Shell
HTTP	Hypertext Transfer Protocol
CIDR	Classless Inter-Domain Routing
I/O	Input or output
SSL	Secure Sockets Layer
AR	Augmented reality
VR	Virtual reality
Networking and Content Delivery	
ELB	Elastic Load Balancing
VPC	Virtual Private Cloud
NAT gateway	Network Address Translation gateway
Compute	
EC2	Amazon Elastic Compute Cloud
AMI	Amazon Machine Image
Storage	
S3	Amazon Simple Storage Service
EFS	Elastic File System
Database	
RDS	Amazon Relational Database Service
Security, Identity, & Compliance	
IAM	Identity and Access Management
WAF	Web Application Firewall

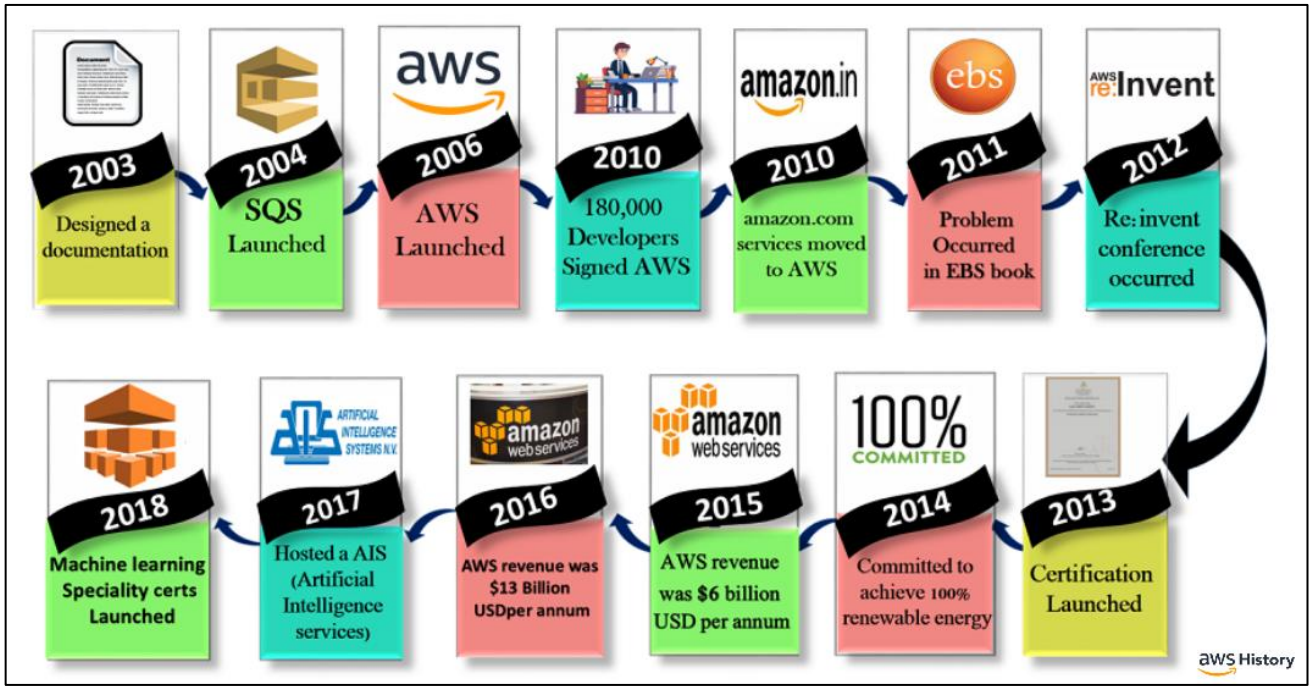
Appendix 2

Appendix 2.1



Appendix 2.1: Charts showing rising demand for online learning [1]

Appendix 2.2



Appendix 2.2 History Timeline of AWS

Appendix 2.3

Appendix 2.3 AWS Global Infrastructure [5]