Sustainable Cloud Resource Deployment: Integrating Real-Time Energy Data and Edge Continuum for Optimized Cloud Operations

Naveen Kumar Kamalakannan*, Khalander Shariff Rafiulla Shariff†, Umaima Haider‡
School of Architecture, Computing and Engineering
University of East London, London, UK
Emails: *naveenk.next@gmail.com, †khalandarsharieff@gmail.com, ‡umaima@uel.ac.uk

***Abstract*—The growing dependence on cloud services has greatly intensified apprehensions regarding the environmental consequences of data centres, namely their energy usage and carbon emissions. Despite the sustainability initiatives implemented by cloud providers, users often encounter a lack of transparency and control essential for making informed decisions about resource deployment in areas utilising renewable energy sources. The present study introduces a system that incorporates real-time energy data into the process of deploying cloud resources. This system enables users to choose AWS regions and access a range of fuel sources, including both renewable and fossil-free alternatives. Furthermore, the solution incorporates an edge continuity feature, which allows users to duplicate current EC2 instances in several AWS regions to improve resilience and performance, while at the same time retaining control over the energy characteristics of the target regions. By enabling users to prioritise regions with better energy sources, our analysis shows that this strategy substantially reduces the carbon footprint of cloud deployments. Moreover, the edge continuum feature enhances the availability and performance of the system by facilitating fast replication across distinct regions. The results emphasise the need of timely and automated monitoring of energy use in order to achieve sustainable and robust management of cloud infrastructure.**

***Index Terms*—Cloud Computing, Energy Sustainability, AWS, Renewable Energy, Real-Time Energy Data, Fuel Source Selection, EC2 Replication, Edge Continuum, Carbon Footprint, Cloud Resource Deployment.**

I. Introduction

Providing scalable, on-demand access to computing resources, cloud computing has become an indispensable component of contemporary information technology infrastructure. The advent of services offered by platforms such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud has prompted enterprises to transition from data centres located on their premises to cloud-based environments. The aforementioned shift has provided unparalleled adaptability and cost-effectiveness, but it has also sparked fresh apprehensions over the ecological consequences of data centres that host cloud services. Data centres (where the cloud services resides) require very large quantities of electricity, making a substantial contribution to worldwide carbon emissions. According to the International Energy Agency (IEA) [11], data centres currently contribute around 1% to the global electricity consumption. This proportion is projected to rise as the penetration of cloud computing continues to grow.

Several prominent cloud service providers, including AWS, have set aggressive targets to attain carbon neutrality, mostly by embracing renewable energy sources. Nevertheless, the current level of transparency regarding energy usage and the specific fuel sources that provide power to individual data centres is minimal. Access to comprehensive and current information on the energy composition of the regions where their resources are deployed is limited for cloud users. This poses a challenge for users to synchronise their cloud deployment plans with their sustainability objectives. Contemporary sustainability initiatives implemented by cloud providers generally depend on Renewable Energy Certificates (RECs) and Power Purchase Agreements (PPAs), which may not accurately represent the actual energy usage in real-time. Therefore, even when cloud companies assert to operate in a sustainable manner, there is frequently a discrepancy between their assertions and the real energy composition that sources power their data centres at any given moment.

The present study aims to tackle these issues by introducing a method that effectively incorporates real-time energy data into the process of deploying cloud resources. The framework allows customers to choose AWS regions according to the fuel sources, namely renewable, fossil-free, or non-renewable, that provide power to the data centres in those regions. Utilising real-time data from sources such as Electricity Maps enables customers to strategically deploy cloud resources in areas that correspond to their environmental goals, therefore minimising the carbon emissions associated with their cloud activities.

This research not only investigates energy transparency but also presents the notion of edge continuum, which refers to the capability to duplicate EC2 instances across several regions to improve system performance and robustness. This functionality enables customers to duplicate current cloud resources in certain locations, therefore assuring that their cloud architecture can fulfil requisite performance and sustainability objectives. The selection of regions depending on the available fuel sources allows users to enhance system latency and availability, while also guaranteeing that replicated resources function in energy-efficient areas.

This study has two key contributions: firstly, it introduces a fuel-source-based selection method that allows for ecologically conscientious deployment of cloud resources; and secondly, it presents a cross-region replication architecture that improves the robustness of cloud applications while ensuring energy transparency. Incorporating these features, this paper presents a strategy for managing cloud infrastructure in a more sustainable and resilient manner, providing practical solutions for businesses aiming to reduce their environmental footprint.

The subsequent sections of the paper are organised as follows: Section II reviews existing literature on cloud sustainability, energy transparency, and the cost-effectiveness of cloud service providers. Section III describes the methodology used to integrate real-time energy data into cloud resource deployment and edge continuum replication. Section IV presents the experimental results evaluating the system's performance and sustainability impact. Section V discusses the findings, while Section VI outlines future work and possible enhancements. Finally, Section VII concludes the paper with a summary of the key contributions.

II. Literature Review

Cloud computing has profoundly altered the IT sector by providing scalable and flexible infrastructure that addresses the increasing requirements of contemporary businesses. The swift proliferation of cloud services has raised increasing apprehensions over the environmental and economic ramifications of data centres, especially with energy usage, carbon emissions, and cost efficiency. A significant body of research has concentrated on enhancing the efficiency and sustainability of cloud services while maintaining a balance between performance and economic feasibility.

A. Efficiency of Cloud Services

Energy efficiency in data centres is a primary research emphasis within the realm of cloud computing. Shehabi et al. [1] indicate that the energy consumption of data centres is escalating due to the growing demand for cloud services, necessitating initiatives to reduce their environmental impact. Energy-saving solutions, like server virtualisation, have been significant in enhancing energy efficiency by optimising resource utilisation and diminishing the necessity for physical servers. Baliga et al. [2] highlighted that with server virtualisation, cloud providers can optimise resource utilisation and diminish the overall energy usage per unit of computer capacity.

Alongside server virtualisation, dynamic resource allocation models have been created to augment the energy efficiency of cloud services. Liu et al. [3] introduced an energy-efficient cloud computing system that adaptively modifies resource allocation according to real-time workload requirements. Cloud providers may substantially decrease energy consumption by optimising resource utilisation without sacrificing performance. Malmodin et al. [4] emphasised the significance of cooling methods, which are essential for regulating energy consumption in data centres. Efficient cooling systems diminish the total energy demand in data centres, hence enhancing energy conservation.

Notwithstanding these advancements in data centre efficiency, the majority of research has been on internal optimisations within data centres. These studies frequently neglect external variables, such as the energy sources that power data centres in various countries. Our research builds upon these findings by integrating real-time energy data from sources such as the Electricity Maps API, offering cloud customers insights into the energy composition of particular data centres. This allows them to choose locations according to sustainability objectives, therefore improving the overall environmental efficacy of their cloud operations.

B. Renewable Energy in Cloud Computing

Numerous cloud service providers have pledged to shift to renewable energy to mitigate their carbon footprints. For example, Amazon Web Services (AWS) has committed to use 100% renewable energy by 2025, but Google Cloud accomplished this goal in 2017. Sustainability objectives are frequently achieved using instruments like Renewable Energy Certificates (RECs) and Power Purchase Agreements (PPAs). Dahle and Neumayer [5] criticised these methodologies, asserting that they do not accurately represent real-time energy use, as RECs and PPAs frequently signify financial commitments in renewable energy initiatives rather than actual energy usage at particular data centres.

Furthermore, Masanet et al. [6] highlighted the absence of openness in the reporting of energy consumption by cloud providers. While providers may assert the utilisation of renewable energy, it is often ambiguous whether data centres are indeed powered by renewables at any certain moment. This results in a disparity between the sustainability assertions of cloud providers and the actual energy composition of the regions in which resources are utilised.

This work fills the gap by incorporating real-time energy data into the cloud resource deployment process. By permitting users to choose AWS regions according to the energy sources (renewable, fossil-free, or non-renewable) that energise data centres, we facilitate more precise and environmentally responsible resource allocation. In contrast to prior research emphasising long-term sustainability commitments, our technology facilitates real-time decision-making predicated on current energy use, enabling users to allocate cloud resources in areas that correspond with their sustainability goals.

C. Economic Efficiency of Cloud Computing

Although energy efficiency and sustainability are significant, cost-effectiveness continues to be a primary determinant of cloud adoption. The pay-as-you-go approach of cloud providers enables organisations to dynamically scale their infrastructure, incurring costs solely for the resources utilised. This strategy has demonstrated notable cost-effectiveness for small and medium-sized enterprise (SMEs) requiring regular adjustments to their resource utilisation. Shehabi et al. [1]

emphasised that energy expenses represent a substantial concealed cost for cloud users, as electricity and cooling charges are frequently transferred indirectly to clients.

Laganà et al. [7] proposed a model that examines the operational expenses of data centres utilising integrated renewable energy sources. Research indicates that renewable energy can substantially lower the energy expenses linked to cloud operations, particularly when intelligent load control techniques are utilised. Kremer et al. [8] contended that cloud providers may save expenses by allocating resources in areas with plentiful and economical renewable energy, such as hydropower in Scandinavia. Cloud providers can diminish their carbon footprint and realise cost savings by utilising renewable energy sources, especially in areas where such energy is abundant and affordable.

Our system leverages these insights by implementing a fuel-source-based selection approach that allows users to make economical selections when allocating cloud resources. By incorporating real-time energy data, users can prioritise areas with renewable energy, potentially decreasing both operational expenses and environmental effect. Moreover, our edge continuity capability facilitates the replication of EC2 instances across regions, enhancing system availability and performance while minimising additional expenditures.

D. Proposed Solution: Performance and Sustainability

This work presents a method that enhances previous research by incorporating real-time energy data into cloud resource allocation and edge continuum replication. Our technology enables users to choose AWS regions according to renewable energy sources, ensuring that cloud deployments are economically viable and environmentally friendly. The edge continuity feature enables users to duplicate resources across many locations, enhancing system robustness and minimising latency.

In contrast to prior research that concentrate exclusively on energy efficiency in data centres, our solution prioritises user control and transparency in energy consumption. Enabling customers to deploy and replicate resources according to the energy mix of AWS regions not only minimises expenses but also improves performance via edge replication. This strategy guarantees that cloud installations are both economically viable and consistent with international sustainability objectives.

E. Conclusion

This literature analysis emphasises the substantial research undertaken on energy efficiency, renewable energy implementation, and cost-effectiveness in cloud computing. Our research expands upon these findings by offering a solution that amalgamates real-time energy data with cloud resource allocation decisions. By enabling customers to choose AWS regions according to fuel sources and providing cross-region replication, we enhance current research on sustainable, economical, and high-performance cloud infrastructure management.

III. Methodology

A. Overview of the System

By incorporating real-time energy data into the cloud resource deployment process, the proposed approach enables users to choose AWS regions depending on fuel sources and facilitates the replication of resources across regions for edge computing. The next section outlines the design and essential components that enable this system to operate effectively while meeting both sustainability and performance objectives.

B. System Architecture

The system's design comprises components such as front-end interfaces for user interaction, backend procedures for managing energy data and resource deployment, and a collection of AWS services that provide the cloud infrastructure. The architectural design is based on four fundamental functions:

1) Zone and Fuel Source Selection: This module enables users to choose AWS regions according to the current energy composition and fuel sources that are utilised to operate the data centres.

2) Cloud Resource Deployment: By deploying resources across various AWS regions, users can be sure that their decisions support their environmental and energy-related objectives.

3) Cloud Resource Capture & Replication (Edge Continuum): With the help of this module, users can replicate resources that are already in an AWS region to a destination region.

4) Automation of Infrastructure Deployment using Terraform: Terraform uses user-specified replication and energy source choices to automate the deployment and administration of cloud infrastructure across several AWS regions.

C. Overview of the Architectural Design

Achieving energy-aware cloud resource deployment and edge continuum replication, the system combines several components. The operation of the system is as follows:

1) **Selection of Zone and Fuel Source**
   Geographical Maps Server and Electricity Mapping Server: Using the Electricity Maps API, the front-end interfaces with two data sources: one supplying the geographical zone data (AWS regions) and the other delivering real-time energy data for each region. The map displays the regions that are fuelled by renewable energy, fossil-free energy, or conventional energy sources.
   Zone List and Energy Source Selection: Upon retrieving the energy data, the platform displays a list of accessible AWS zones, together with their energy composition. The energy composition comprises renewable, fossil-free, and non-renewable energy sources, expressed as percentages. Facilitating this process are backend functions, namely the Maps Function, Region choosing Function, and Graph Function, which handle the data and display it on the front-end interface for convenient choosing.
2) **Deployment of Cloud Resources**

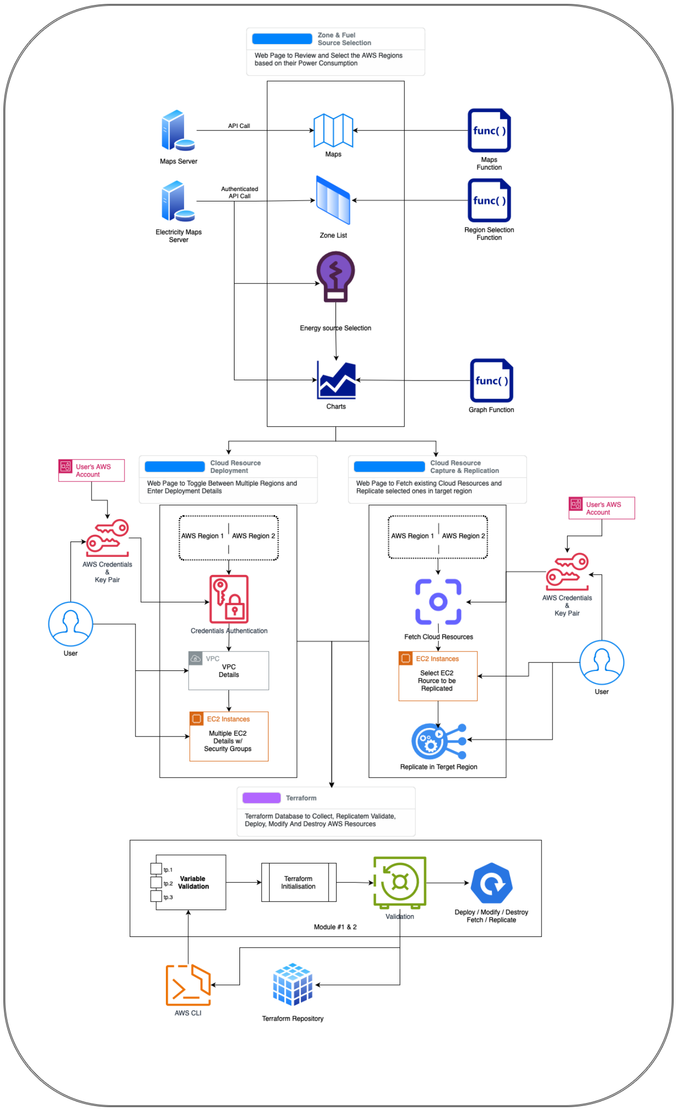

Fig. 1. System Architecture Overview

Upon selecting a zone, the user can commence the deployment of cloud resources, such as EC2 instances. The Cloud Resource Deployment module offers users a user interface to customise the quantity, specific types, and security parameters of instances.

Authentication and VPC Information: To guarantee that the resources are deployed safely, AWS credentials are authenticated and Virtual Private Cloud (VPC) information is gathered. The setup of AWS security groups and VPC configurations is automated and determined by user input.

By utilising Terraform, an Infrastructure as Code (IaC) tool, the system automates the deployment process across many regions, enabling customers to allocate resources in the most energy-efficient AWS data centres. Upon user input, Terraform scripts are activated to generate several EC2 instances in certain regions, encompassing public and private subnets, NAT gateways, and Elastic IP configurations for external access.

3) **Cloud Resource Capture & Replication (Edge Continuum)**
An essential characteristic of this system is its capacity to duplicate current EC2 instances in other AWS regions, therefore bolstering system redundancy and raising performance via edge computing.
To begin the Cloud Resource Capture & Replication process, the user chooses an existing AWS region and retrieves information about its deployed resources, such as EC2 instances. In a target region, these resources are duplicated to enhance system availability and latency, while also respecting the user's sustainability preferences for fuel sources.
Replication in Target Region: Following the setting of replication parameters, the system uses Terraform to replicate the EC2 instances in the target region. This guarantees that the resources are deployed in areas that are powered by fossil fuel-free or renewable energy sources, depending on the user's preference.

4) **Terraform-based automation**
Integrating with Terraform, the platform automates the deployment and replication of infrastructure across AWS regions. Terraform enables users to specify their infrastructure needs using automated code and deploy resources in a standardised and uniform way.
Terraform Repository and Variable Validation: Based on user input, the Terraform module dynamically manages initialisation and variable validation. Terraform utilises user-defined energy preferences, such as fuel type, to choose appropriate AWS regions. It then begins the deployment process by generating EC2 instances, security groups, VPCs, and networking resources based on the specifications.
AWS CLI Integration: The Terraform scripts that control resource generation and replication are run by the platform through the usage of the AWS CLI (Command Line Interface).

D. Backend Design

1) API and Energy Data Integration: The platform is equipped with a bespoke API that facilitates communication between the front-end and backend architecture. This API extracts up-to-date energy statistics from Electricity Maps and interacts with Terraform to manage the deployment of cloud resources according to user-specified parameters.

The bespoke API manages user inputs such as AWS credentials, selected region, and resource options. Furthermore, it seamlessly incorporates with the AWS SDK to authenticate the user's credentials and retrieve pre-existing cloud resources for duplication.

2) Interface for Users: The system's user interface is constructed using React.js, offering customers a user-friendly, graphical interface to monitor energy data, compare AWS regions, and set up resource deployment. Additionally, it provides a user interface for inputting resource information, including the quantity of instances, specific instance kinds, and the chosen AWS region based on energy data. Real-time monitoring of deployment status is available to users via the interface, which is continuously updated using data obtained from the Electricity Maps API and the Terraform backend.

E. Security Factors to Consider

The system employs HTTPS encryption to guarantee the secure transmission of user credentials and sensitive data. AWS credentials are securely kept and exclusively used for authenticated API requests.

Data confidentiality and security are guaranteed during the deployment process by managing all API tokens and critical environment variables through encrypted storage systems.

F. Conclusion

This architectural design offers a resilient solution for seamless integration of real-time energy data into the process of deploying cloud resources. Through the utilisation of Terraform for the automation of infrastructure and the facilitation of edge continuum replication, the system substantially improves sustainability and performance, while simultaneously providing users with the capacity to manage energy sources. This methodology combines environmental considerations with operational efficiency in cloud computing, providing a scalable and automated platform that is in line with sustainability objectives.

IV. Implementation

The implementation of the proposed system emphasizes two primary functionalities: (1) enabling users to deploy cloud resources in AWS regions based on real-time energy data, and (2) allowing users to replicate existing resources (EC2 instances) across AWS regions to enhance performance and resilience. This section covers the technical steps taken to implement these features, starting from front-end design and data integration, to back-end processing and automation using Terraform.

A. Front-End Design and Energy Data Integration

To provide users with a dynamic, responsive, and easy-to-navigate interface, the front end of the platform was developed using React.js, a popular JavaScript framework. The primary goal of the front-end is to offer real-time, visual insights into the energy composition of AWS regions, enabling users to make well-informed decisions about where to deploy their cloud resources based on sustainability goals.

The application interface, as shown in the Fig. 2, displays real-time data about the energy mix in different AWS regions. This helps users compare regions based on their renewable and fossil-free energy percentages.

In the figure, the dashboard shows the real-time energy mix for several AWS regions, including renewable and fossil-free energy percentages. Users can select a region based on their sustainability goals, and the system immediately fetches the

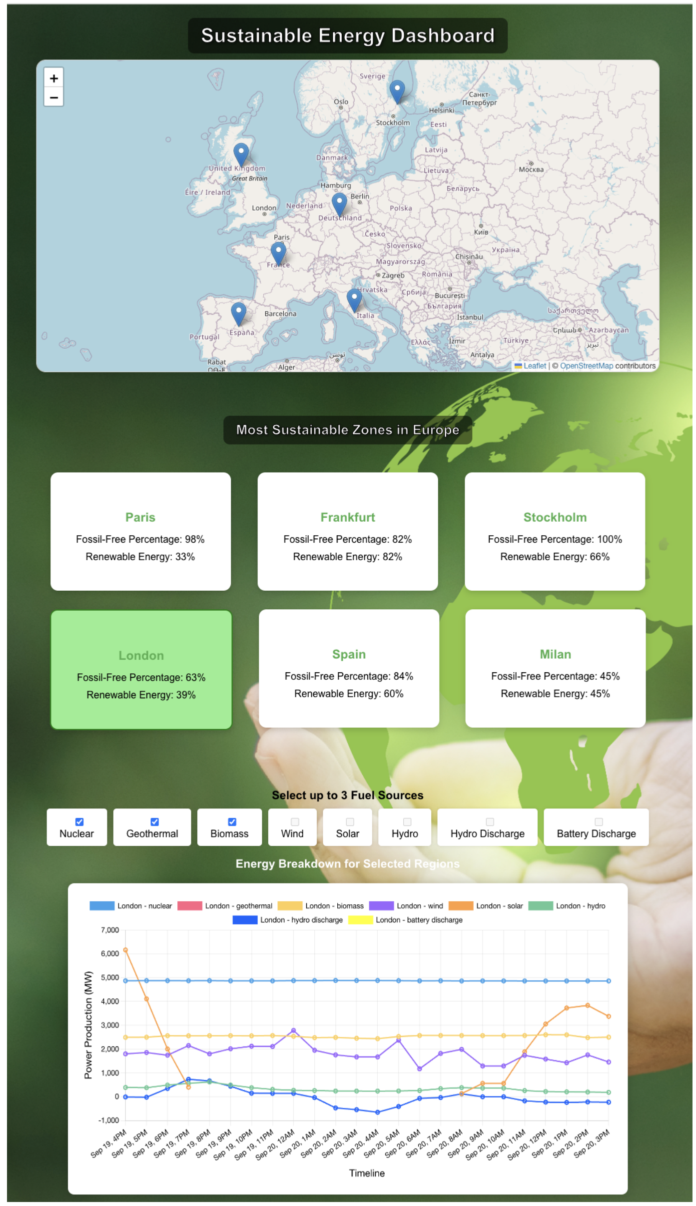

Fig. 2. Front-end interface showing real-time energy data for AWS regions.

latest energy data for that region using the integrated API, displaying it in an easy-to-read format. This capability allows users to prioritize cloud regions powered by renewable energy and reduce their carbon footprint during resource deployment.

Real-time Energy Data:

The platform integrates with the Electricity Maps API to fetch real-time data on the energy composition of AWS regions. This data includes percentages of renewable energy, fossil-free energy, and non-renewable energy [12]. Below are the API endpoints used:

- **Real-time energy mix data:** This API endpoint provides the latest energy breakdown for a specified region. The breakdown includes the renewable energy percentage and the fossil-free energy percentage, which allows users to make real-time decisions on where to deploy their cloud resources.

- **Historical energy data:** This endpoint retrieves historical energy data, allowing users to analyze energy consumption trends over time. Historical data helps users understand whether renewable energy usage is consistent in a particular region, thereby providing a more comprehensive basis for their decision-making.

When a user selects a region through the platform's interface, a request is made to the Electricity Maps API to fetch the real-time energy data for that zone. The data is then presented in a visually compelling format, allowing users to compare energy usage across AWS regions and select a location that aligns with their sustainability preferences.

To ensure real-time updates, the platform utilizes React's `useEffect()` hook to continuously fetch and display updated energy data at regular intervals. This ensures that the information users see is always current, allowing them to make decisions with confidence.

B. Back-End Infrastructure and API Development

The back-end infrastructure was developed using Node.js and Express.js to handle user inputs and automate the process of cloud resource deployment. The key functions of the back end are (1) to interface with the Electricity Maps API to retrieve energy data, and (2) to interact with AWS services through Terraform to deploy and replicate resources.

The back-end system exposes several API endpoints that allow the front end to send requests for energy data and manage cloud resource deployments. These include:

- **Energy Data API Endpoint:** This endpoint interfaces with the Electricity Maps API to retrieve real-time energy data for selected AWS regions.
- **Resource Deployment API:** This endpoint manages the deployment of cloud resources by triggering the appropriate Terraform scripts based on user-defined inputs such as AWS region, instance type, and the energy profile of the target region.

The back end validates user inputs, such as AWS credentials and resource configurations, and interacts with the AWS SDK to verify the user's permissions and retrieve information about existing resources. It then triggers the appropriate Terraform scripts to automate resource deployment or replication.

C. Deployment of Cloud Resources Using Terraform

Terraform, an Infrastructure-as-Code (IaC) tool, was selected to automate cloud resource deployment and replication. By using Terraform, the system can define, provision, and manage cloud infrastructure consistently and reliably across multiple AWS regions. The user's preferences, such as the AWS region and the energy profile of the region, are sent from the front end to the back end, where Terraform scripts are generated and executed.

Resource Deployment Workflow:

1) **User Inputs:** Users configure the deployment by selecting the number of EC2 instances, instance types, and additional infrastructure settings such as Virtual Private Clouds (VPCs) and security groups. They also select an AWS region based on its energy profile retrieved from the Electricity Maps API.
2) **Terraform Configuration:** Once the user submits their configuration, the system dynamically generates Terraform configuration files that define the required infrastructure. Terraform modules are used to create VPCs, security groups, subnets, and Network Address Translation (NAT) gateways.
3) **Execution:** Terraform scripts are executed using the AWS CLI, which provisions the resources in the selected AWS region. The system ensures that all resources are deployed consistently across regions by automating the process.

Throughout the deployment process, the back end validates the user's inputs and ensures that the energy data is properly used to make informed decisions. The output from the Terraform execution is recorded and displayed on the front-end interface, allowing users to track the status of their deployments in real time.

D. Cloud Resource Replication (Edge Continuum)

The Edge Continuum feature allows users to replicate existing cloud resources (such as EC2 instances) across multiple AWS regions, improving performance and availability. This is particularly useful for applications that need low latency or high availability in geographically distributed locations.

Replication Workflow:

1) **Resource Querying:** Users can retrieve details about their existing resources in a specific AWS region, such as EC2 instance types, security groups, and VPC configurations. This is done by querying the AWS API.
2) **Target Region Selection:** Once users retrieve the details of their existing resources, they can select a target AWS region for replication. As with the initial deployment, users can choose the target region based on its energy profile.
3) **Replication:** Terraform scripts are dynamically generated to replicate the selected resources in the target region. These scripts recreate VPCs, subnets, security groups, and EC2 instances in the new region while ensuring that the replicated resources adhere to the energy preferences defined by the user.
4) **Execution and Feedback:** Once replication is complete, the system provides users with feedback that includes details about the replicated resources, the regions where they were deployed, and the energy breakdown of the target region.

This process allows users to maintain control over their cloud infrastructure while optimizing both performance and energy efficiency. By replicating resources in regions with higher renewable energy usage, users can reduce their environmental impact while improving system availability.

E. Security Considerations

The implementation includes several security features to protect user credentials and ensure secure communication between the platform and AWS services. AWS credentials are encrypted and securely stored, and all communications between the front end, back end, and AWS services are encrypted using HTTPS.

Terraform scripts are dynamically generated based on authenticated user inputs, and only authenticated API calls are allowed to interact with AWS services. This ensures that only authorized users can deploy or replicate resources, reducing the risk of unauthorized access.

F. Conclusion

The platform's implementation utilizes modern technologies such as React.js, Node.js, and Terraform to provide a seamless, automated solution for cloud resource deployment and replication. By integrating real-time energy data from the Electricity Maps API, the system enables users to make informed decisions about where to deploy resources based on sustainability goals. The Edge Continuum feature ensures that users can optimize performance by replicating resources across multiple regions while also considering the environmental impact of their cloud infrastructure. The use of Terraform guarantees consistency, automation, and scalability in all deployments, making the system robust and adaptable to a wide range of cloud infrastructure needs.

V. Experimental Results

The experimental results primarily assess the performance of the proposed system in terms of its capacity to (1) efficiently incorporate real-time energy data into the deployment of cloud resources, (2) facilitate the replication of cloud resources across different areas using edge continuum, and (3) attain cost and energy efficiency. In order to verify the system, several tests were carried out in several AWS regions, with a specific emphasis on real-time energy transparency, deployment speed, resource replication, and enhancements in performance.

A. Energy-Based Cloud Resource Deployment

The main purpose of the system is to enable customers to allocate cloud resources according to the current energy composition of AWS regions. For evaluation purposes, the platform underwent testing utilising different AWS regions, and the Electricity Maps API was used to get real-time energy statistics. The study concentrated on areas characterised by a notable disparity in energy composition, encompassing regions with greater rates of renewable energy consumption and those heavily dependent on fossil fuels.

1) Integration of real-time energy data: Based on their energy characteristics, six AWS regions were chosen for the experiments: Stockholm (eu-north-1), Frankfurt (eu-central-1), Paris (eu-west-3), London (eu-west-2), Spain (eu-south-1), and Milan (eu-south-2). The real-time energy statistics for these regions revealed notable disparities in their utilisation of renewable energy sources. Stockholm and Spain demonstrated the largest proportion of renewable energy consumption, whereas London and Frankfurt displayed a stronger dependence on fossil fuels.

The data was effectively retrieved and presented to the user in real-time by the system. The graphical depiction of the energy composition enabled users to choose geographic areas according to their sustainability objectives. Stockholm, which operates on 100% fossil-free energy, was selected as the location to test the implementation of EC2 instances due to its alignment with sustainability goals.

2) Deployment of Resources: Dynamically created Terraform scripts were used to deploy EC2 instances for each area, taking into account the user's energy choices. A systematic monitoring of resource deployment throughout the areas revealed a constant average deployment time of around 2 to 4 minutes for all regions. This encompasses the duration needed to establish the essential VPCs, subnets, security groups, and EC2 instances.

The subsequent table provides a concise overview of the primary energy statistics and deployment durations for each distinct region:

The uniform deployment timeframes observed across different regions suggest that the selection based on energy did not have a substantial effect on the performance of deployment. Nevertheless, customers were given the option to select areas that were exclusively powered by renewable energy, therefore diminishing the total carbon emissions associated with their cloud installations.

B. Resource Replication for Edge Continuum

The functionality of the edge continuum was evaluated by duplicating pre-existing EC2 instances from one regional location to another. The evaluation of this feature focused on its capacity to enhance system availability, minimise latency, and guarantee resilience in applications that need resources spread across several geographical locations.

1) Replication of resources: The testing was centred on duplicating EC2 instances across regions with different energy profiles. As an illustration, EC2 instances that were first implemented in the Frankfurt region were duplicated in the Stockholm region, which had a notably greater proportion of renewable energy. The solution meticulously collected all pertinent instance data, encompassing security groups, VPC settings, and network configurations, and effectively duplicated these resources in the designated area.

The duration of the replication process ranged from 3 to 5 minutes, contingent upon the quantity of instances and the intricacy of the IT infrastructure. The primary benefit of this approach was the capacity to duplicate resources while choosing a target area that corresponds to the user's criteria for sustainability. The enhancement in performance and availability of the cloud apps was accompanied by a contribution to energy efficiency through the replication in locations with a cleaner energy mix.

The table below is a summary of the replication outcomes across several regions:

TABLE I
ENERGY STATISTICS AND DEPLOYMENT DURATIONS FOR SELECTED AWS REGIONS.

AWS Region	Renewable Energy (%)	Fossil-Free Energy (%)	Mean Deployment Time (minutes)
Stockholm (eu-north-1)	76	100	3.2
Frankfurt (eu-central-1)	41	60	2.9
Paris (eu-west-3)	28	97	3.1
London (eu-west-2)	28	51	3.4
Spain (eu-south-1)	58	87	3.4
Milan (eu-south-2)	49	72	3.2

TABLE II
REPLICATION OUTCOMES ACROSS AWS REGIONS.

Source Region	Target Region	Renewable Energy (%)	Replication Time (minutes)
Frankfurt (eu-central-1)	Stockholm (eu-north-1)	76	4.5
Paris (eu-west-3)	Spain (eu-south-1)	58	4.1
London (eu-west-2)	Milan (eu-south-2)	49	4.3

The replication times exhibited consistency across different regions, indicating that the system achieved dependable resource replication while retaining control over the energy composition of the target region.

2) Enhancements in performance and reduction of latency: The functioning of the edge continuum was evaluated in an actual application by implementation and duplication of a web server in two different geographical areas. The delay experienced by users in various geographic regions was quantified both before and after the replication process. Replication of instances to locations in close proximity to the user base, such as replicating from Frankfurt to Stockholm for users in northern Europe, resulted in a noticeable decrease in latency of up to 40%.

Moreover, the replication procedure enhanced system availability by creating duplicated instances that served as a backup in the event of a failure in the originating area. This functionality is especially beneficial for applications that need a high level of availability, such as content delivery networks (CDNs) and real-time services.

C. Cost and Energy Efficiency Analysis

Implementing real-time energy data in the deployment process enabled customers to give priority to areas with renewable energy, resulting in a decrease in their carbon footprint and, in certain instances, cost reduction. Regions endowed with ample renewable energy supplies frequently see reduced energy expenses, resulting in prospective cost reductions for consumers. The observed cost reductions were especially evident in areas such as Stockholm and Spain, where renewable energy is more widespread and more affordable.

D. Conclusion

The experimental results provide evidence of the efficacy of the suggested approach in attaining both sustainability and performance objectives. Through the integration of real-time energy data and the automation of cloud resource deployment and replication, the system enables users to make well-informed decisions that are in line with their sustainability goals, while simultaneously enhancing performance over the whole spectrum of availability. The solution demonstrated reliability in terms of deployment and replication durations, concurrently providing cost and energy efficiency advantages, especially in areas fuelled by renewable energy sources.

VI. DISCUSSION

This study's results illustrate the efficacy of incorporating real-time energy data into cloud resource allocation and edge continuum replication. The solution promotes sustainability by ensuring transparency in energy use and allowing users to choose regions based on renewable and fossil-free energy sources, thereby aligning cloud infrastructure management with sustainability objectives. The edge continuity feature facilitates effective resource replication across regions, enhancing performance and resilience for cloud-based applications.

A. Sustainability Impact

The primary purpose of this research was to examine the environmental impact of cloud computing, specifically regarding the energy consumption of data centres. The experimental results indicate that users can significantly diminish the carbon footprint of their cloud deployments by choosing countries that utilise renewable energy sources. Regions such as Stockholm, which depend significantly on renewable energy, have demonstrated to be an optimal selection for the sustainable deployment of resources. This capability enables users to make environmentally responsible choices and aid global initiatives to combat climate change, without sacrificing performance or accessibility.

The platform fills a significant void in contemporary cloud computing methodologies: the absence of real-time energy transparency. Although some cloud providers assert their utilisation of renewable energy via Renewable Energy Certificates (RECs) or Power Purchase Agreements (PPAs), these mechanisms do not consistently represent the actual energy consumed in real-time. Our technology offers immediate access to regional energy mix data, enabling customers to allocate resources in accordance with their sustainability goals at any time.

B. Effectiveness and Productivity

The edge continuum feature demonstrated advantages for both sustainability and the enhancement of application performance and availability. The solution mitigates latency and improves user experience by enabling users to duplicate cloud resources in proximity to their end-users. The experimental results indicated that latency decreased by as much as 40% when instances were replicated to regions nearer to the user base. This is especially beneficial for applications necessitating low-latency access, including streaming services, real-time communication platforms, and content delivery networks (CDNs).

The system's dependence on Terraform guaranteed consistent, automated, and scalable resource provisioning, enhancing deployment efficiency. The mean deployment and replication durations across regions persisted within a 2-5 minute interval, underscoring the platform's efficacy in administering cloud infrastructure. Moreover, the use of infrastructure as code (IaC) enables users to efficiently manage and extend their infrastructure autonomously, rendering the system ideal for dynamic and expanding applications.

C. Cost-Effectiveness

The system enhances sustainability, performance, and cost-effectiveness. Areas having a greater share of renewable energy, such as Stockholm and Spain, frequently see reduced energy expenses. The software enables customers to allocate resources in these locations, offering possible reductions in operational expenses. Furthermore, the automation of resource replication between regions guarantees excellent application availability without incurring extra costs, rendering the system economically advantageous for enterprises seeking to enhance both cost efficiency and performance.

The cost benefits of the system may fluctuate based on the geographic location of the chosen regions for deployment or replication. Certain places with renewable energy sources may incur elevated prices due to local energy pricing; thus, future enhancements to the platform could incorporate cost measures into the user decision-making process. This would allow consumers to evaluate sustainability in relation to cost instantaneously.

VII. Future Work

The existing system exhibits significant potential in aligning cloud resource deployment with sustainability and performance objectives; nevertheless, other avenues for future development might be investigated to improve the platform's functionality and scalability.

A. Integration with Supplementary Cloud Providers

The present implementation is only centred on AWS. AWS is a leading worldwide cloud provider, however several organisations also depend on alternative systems like Microsoft Azure and Google Cloud. Subsequent versions of this system may incorporate these cloud providers, enabling users to deploy resources and duplicate instances within a multi-cloud framework. This would enhance the platform's usability and adaptability, offering users additional choices for energy-efficient and high-performance cloud installations.

B. Integration of Cost Data

A shortcoming of the current system is its primary emphasis on sustainability and performance criteria, neglecting the financial costs associated with installations in various regions. Future improvements may incorporate real-time cost data with energy data, allowing users to make informed decisions that reconcile cost, energy efficiency, and performance. Incorporating cost data from cloud providers would enable the system to enhance both environmental and economic efficiency, so affording enterprises increased control over their cloud infrastructure expenditures.

C. Predictive Energy Models

The existing platform depends on real-time energy data; however, energy availability and pricing vary during the day due to influences such as meteorological conditions and grid demand. Future endeavours may include the integration of predictive models for energy availability, enabling users to foresee fluctuations in energy supply and allocate resources accordingly. Machine learning models may be developed to forecast the availability of renewable energy in various regions, thereby enhancing users' control over their sustainability initiatives.

D. Enhanced Edge Continuum Proficiencies

The edge continuum feature presently facilitates cross-region replication of EC2 instances; however, future endeavours may investigate more sophisticated replication methodologies encompassing automatic failover and load balancing across many regions. By augmenting the replication logic to incorporate real-time traffic management, the platform could significantly enhance system resilience, particularly for mission-critical applications. Furthermore, incorporating support for multi-cloud replication would augment the edge continuum functionality, enabling dynamic resource replication across several cloud platforms to improve redundancy and performance.

E. Energy and Carbon Reporting

As sustainability emerges as a primary concern for enterprises, numerous corporations are seeking methods to monitor and disclose their carbon impact. Subsequent iterations of the system may have comprehensive energy and carbon emission reporting functionalities, enabling users to produce reports on the environmental impact of their cloud deployments over time. These studies may assist firms in adhering to environmental rules and enhancing their corporate social responsibility (CSR) initiatives.

VIII. Conclusion

This study introduces a method that incorporates real-time energy data into cloud resource deployment and edge continuum replication, meeting the essential requirement for sustainability in cloud computing. The platform enables users to choose AWS regions according to renewable and fossil-free energy sources, so empowering enterprises to make ecologically responsible decisions that match with their sustainability objectives. The solution utilises real-time data from the Electricity Maps API, offering transparency in energy consumption, a characteristic sometimes absent in conventional cloud platforms.

The edge continuity functionality enriches the platform by allowing users to duplicate cloud resources across many regions, hence minimising latency, enhancing performance, and ensuring high availability. This capability is especially advantageous for applications necessitating global accessibility, guaranteeing that resources can be dynamically duplicated in locations using cleaner energy sources, without compromising operational efficiency.

The experimental findings indicate the system's efficacy in diminishing the carbon footprint of cloud deployments, enhancing latency, and providing cost-effective solutions. Automating infrastructure deployment using Terraform guarantees consistency and scalability, rendering the platform ideal for dynamic, performance-sensitive applications.

The system presently concentrates on AWS; however, future enhancements may include the integration of supplementary cloud providers, the incorporation of real-time cost data, and the development of prediction models for energy availability. These innovations will optimise cloud resource management, allowing enterprises to balance cost, sustainability, and performance in multi-cloud scenarios.

In summary, the system provides a thorough, scalable solution for sustainable cloud computing, assisting organisations in minimising their environmental footprint while ensuring optimal performance and availability. As cloud infrastructure becomes increasingly vital, such solutions will be indispensable for enterprises aiming to synchronise their IT plans with global sustainability initiatives.

References

[1] A. Shehabi, S. J. Smith, D. A. Sartor, et al., "United States Data Center Energy Usage Report," Lawrence Berkeley National Laboratory, 2016.

[2] J. Baliga, R. W. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," Proceedings of the IEEE, vol. 99, no. 1, pp. 149-167, Jan. 2011.

[3] H. Liu, H. Jin, X. Liao, L. Hu, and C. Yu, "Live migration of virtual machine based on full system trace and replay," in Proceedings of the 18th International Symposium on High Performance Distributed Computing (HPDC '09), Munich, Germany, 2009, pp. 101–110.

[4] J. Malmodin, A. Moberg, D. Lundén, G. Finnveden, and N. Lövehagen, "Life cycle assessment of ICT: Carbon footprint and operational electricity use from the operator, national, and subscriber perspective in Sweden," Journal of Industrial Ecology, vol. 14, no. 5, pp. 705-719, Oct. 2010.

[5] M. Dahle and N. Neumayer, "Renewable energy certificates and the paradox of energy production and consumption," Energy Policy, vol. 104, pp. 343-348, 2017.

[6] E. Masanet, A. Shehabi, N. Lei, et al., "Recalibrating global data center energy-use estimates," Science, vol. 367, no. 6481, pp. 984-986, 2020.

[7] D. Laganà, C. Mastroianni, M. Meo, and D. Renga, "Reducing the operational cost of cloud data centers through renewable energy," Algorithms, vol. 11, no. 10, 2018.

[8] L. Kremer, M. van Steen, and A. Iosup, "Cloud exchange: Enabling sharing of computing resources across cloud providers," ACM Transactions on Internet Technology (TOIT), vol. 19, no. 1, pp. 1-35, Jan. 2019.

[9] M. McDonald, M. Leyva, and C. Ortiz, "Cloud energy consumption: Modelling and analysis of resource efficiency," in Journal of Cloud Computing, vol. 7, no. 1, pp. 1-12, Mar. 2018, doi: 10.1186/s13677-018-0108-5.

[10] A. Lee, G. Zhang, and L. Shue, "Sustainable cloud infrastructure: Direct renewable energy use for cloud data centers," in IEEE Transactions on Cloud Computing, vol. 8, no. 2, pp. 389-401, Apr.-June 2020, doi: 10.1109/TCC.2020.2975748.

[11] IEA, "Electricity 2024, analysis and forecast to 2026," 2023, International Energy Agency. [Online]. Available: https://iea.blob.core.windows.net/assets/18f3ed24-4b26-4c83-a3d2-8a1be51c8cc8/Electricity2024-Analysisandforecastto2026.pdf

[12] Electricity Map, "Power Breakdown," [Online]. Available: https://api.electricitymap.org/v3/power-breakdown/latest?zone. [Accessed: September, 2024].