CSCI 5902 Term Project – Fall

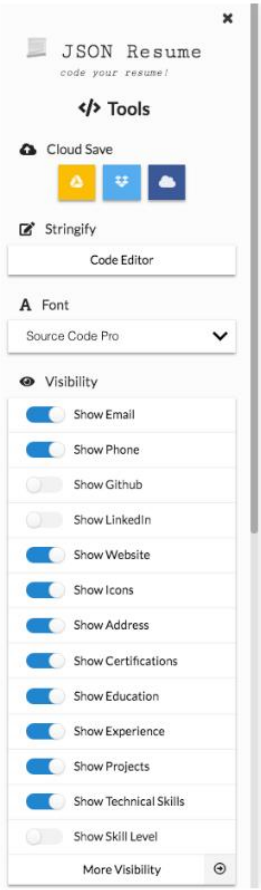
Yogish Honnadevipura Gopalakrishna

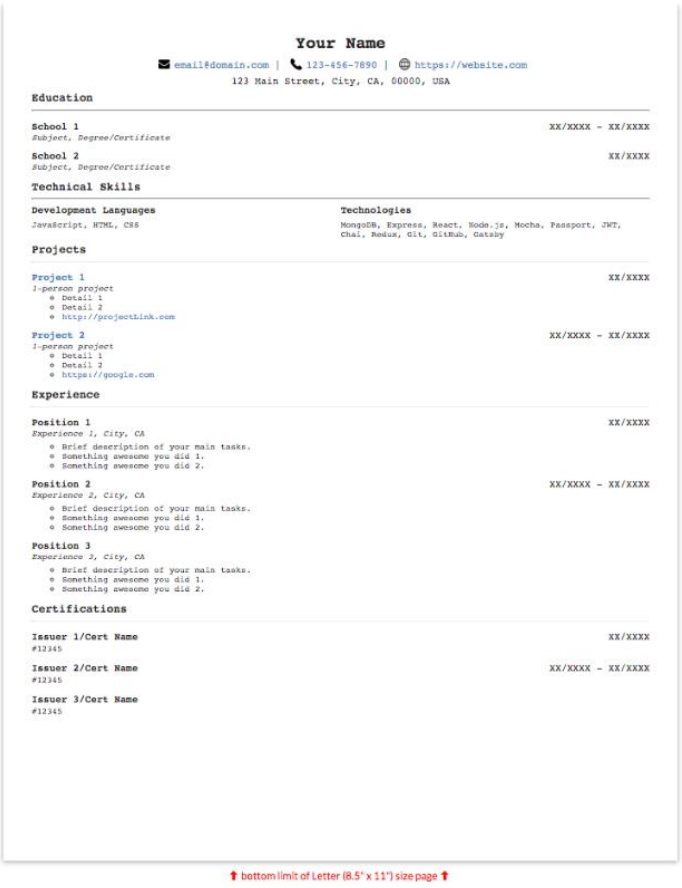
BANNER ID – B00928029

Term Project Report

What is the project about and the reason for choosing this project

My Application github link - https://github.com/aamay001/react-resume
Project Deployment URL - http://34.206.108.31:3000/





This Project is about creating a template for resume creation which uses React for the frontend and uses a JSON file for the backend. The values changed in the backend will be reflected in the frontend and displays it to the user. Transforming the React app with a resume template into a cloud-based project offered an invaluable opportunity to me to exhibit my architectural skills within the context of traditional web application deployment. This project provides hands-on experience with cloud computing, focusing on more conventional approaches to frontend development, backend development and server hosting.

By opting for EC2 to host the frontend and backend, I got to work on Infrastructure as a Service (IaaS) on AWS. This approach allows for more granular control over the server environment, demonstrating the student's understanding of server provisioning and configuration management. It's an opportunity to showcase skills in setting up and managing a web server, handling the backend logic, and connecting it to other AWS services. I used DynamoDb for storing the JSON file which communicates with the backeend.

For system architecture, I integrated AWS Elastic Load Balancing (ELB) to distribute incoming traffic across multiple EC2 instances for enhanced availability and fault tolerance. This adds a layer of resilience to the overall application, building robust and fault-tolerant systems.

Compute

I am designing an architecture for hosting a React frontend and a backend on EC2 instances. Employing Amazon EC2 for both the front-end and back-end components of our resume creator app aligns with the project's requirements and offers a range of advantages.

Cost-Effectiveness:

EC2 provides a pay-as-you-go pricing model, allowing you to pay only for the compute capacity you consume. This aligns well with the variable and unpredictable nature of web applications, ensuring cost efficiency. Reserved Instances and Savings Plans further offer cost savings for predictable workloads with reserved capacity commitments.

Performance Optimization:

EC2 instances come in various types optimized for different use cases (e.g., compute-optimized, memory-optimized). This allows you to choose instances tailored to the specific needs of the frontend and backend components, optimizing performance.

Placement Groups in EC2 enable you to control the placement of instances to achieve low-latency communication, crucial for enhancing application performance.

Security Measures:

EC2 instances integrate with AWS Identity and Access Management (IAM) for fine-grained access control. You can define roles and policies to ensure the principle of least privilege. Virtual Private Cloud (VPC) allows you to isolate resources, control inbound and outbound traffic, and configure network ACLs and security groups to enhance the security posture of the application.

Scalability and Flexibility:

EC2 instances support both vertical and horizontal scaling. Auto Scaling Groups enable automatic scaling of instances based on demand, ensuring that the application can handle varying workloads efficiently.

Load Balancers can be easily integrated to distribute traffic across multiple EC2 instances, improving availability and responsiveness.

Elastic Load Balancing (ELB):

ELB integrates seamlessly with EC2 instances, distributing incoming application traffic across multiple instances to ensure optimal resource utilization and fault tolerance. Application Load Balancers, part of ELB, provide advanced routing and allow for efficient handling of HTTP/HTTPS traffic, enhancing the application's scalability.

Managed Services Integration:

While EC2 provides full control over the infrastructure, you can also integrate with managed AWS services like Amazon RDS for database management, Amazon S3 for object storage, and AWS Lambda for serverless functionality, further optimizing the application architecture.

Explain why the chosen service was the best fit for your application, considering factors like cost, performance, security, and scalability.

Storage

For this project, Amazon S3 and Elastic Block Store (EBS) emerge as the optimal storage solutions for several compelling reasons. Firstly, Amazon S3, known for its scalability and durability, provides an ideal repository for storing profile images. With its seamless integration with content delivery networks (CDN) like Amazon CloudFront, we can efficiently deliver images to users globally, ensuring low latency and enhanced user experience. The versioning and lifecycle management features of S3 enable us to maintain a version history of profile images and automate the transition to cost-effective storage classes as images age, contributing to efficient resource utilization and cost savings.

Elastic Block Store (EBS) is a natural fit for the storage needs of our EC2 instances. EBS volumes provide persistent and high-performance block-level storage, offering the necessary reliability for the backend infrastructure of our application. By choosing appropriate EBS volume types based on workload characteristics, we can tailor the storage performance to meet the demands of our application. The ability to create snapshots of EBS volumes ensures data durability and provides a straightforward mechanism for backups and disaster recovery, aligning with the reliability requirements of our project.

Amazon S3:

Scalability and Durability:

S3 is highly scalable, allowing the application to seamlessly handle an increasing number of profile images. Its distributed architecture ensures durability, reducing the risk of data loss. The ability to scale horizontally and accommodate virtually unlimited data makes S3 suitable for the storage of a large volume of profile images.

Global Content Delivery:

Integration with Amazon CloudFront enhances the global distribution of profile images, reducing latency and ensuring a responsive user experience across diverse geographic locations. CloudFront's CDN capabilities further optimize image delivery by caching content at edge locations, minimizing the load on the origin server.

Versioning and Lifecycle Management:

S3's versioning capability allows for the tracking of changes to profile images over time. This is crucial for maintaining a version history and recovering from unintended changes. Lifecycle policies enable the automated transition of older images to cost-effective storage classes, optimizing costs without compromising accessibility.

Security Features:

S3 supports server-side encryption, ensuring the confidentiality and integrity of stored profile images. This is vital for protecting sensitive user data. Fine-grained access control through IAM policies allows for the implementation of least privilege access, enhancing the overall security posture.

Elastic Block Store (EBS):

Persistent and High-Performance Storage:

EBS volumes provide persistent block-level storage for EC2 instances, ensuring data consistency and reliability. This makes EBS a natural fit for the backend infrastructure of the resume creator application. By selecting appropriate EBS volume types based on workload characteristics, you can tailor storage performance to meet the specific demands of both frontend and backend components.

Snapshots for Data Durability and Backup:

EBS allows for the creation of snapshots, providing a reliable mechanism for data durability, backup, and disaster recovery. Snapshots can be used to restore volumes or create new volumes in case of failures. The ability to automate snapshot creation and management simplifies the backup process, contributing to the overall reliability of the storage infrastructure.

Cost-Effective Scaling:

EBS allows you to scale storage capacity based on the needs of the application. You can dynamically adjust the size of EBS volumes, optimizing costs by only paying for the storage capacity you consume.

Integration with EC2 Instances:

As a block storage solution, EBS seamlessly integrates with EC2 instances, providing low-latency access to data. This integration ensures optimal performance for the backend infrastructure. By leveraging Amazon S3 and Elastic Block Store for storage needs, the resume creator application benefits from scalable, durable, and cost-effective solutions that align with the specific requirements of the project. The combination of S3 and EBS addresses key considerations, including performance, security, reliability, and scalability.

Database

DynamoDB is the best option for database in this project because of its ability to seamlessly scale horizontally makes it an optimal choice for our resume creator app. As the data grow, DynamoDB can effortlessly handle increased read and write throughput. The automatic sharding of data across multiple nodes ensures consistently low-latency access, providing a responsive user experience.

In the dynamic environment of a resume creator app, where user profiles and resume structures may evolve, DynamoDB's flexible schema design proves advantageous. The NoSQL nature of DynamoDB allows us to adapt the database schema to changing requirements without necessitating a major overhaul. This flexibility aligns with the AWS Well-Architected Framework's pillar of Reliability by facilitating easy updates without compromising the availability of our application. Here are some additional advantages of using DynamoDB

Horizontal Scalability:

DynamoDB excels in seamlessly scaling horizontally to handle increased data volumes and traffic. This makes it well-suited for the dynamic and potentially unpredictable growth of user profiles and resume data in the resume creator app. The automatic sharding of data across multiple nodes

ensures that as the application's user base expands, DynamoDB can effortlessly manage higher read and write throughput while maintaining consistently low-latency access.

Cost-Effective Scaling:

With DynamoDB's pay-per-request pricing model, you only pay for the read and write capacity that your application consumes. This provides a cost-effective solution, especially in scenarios where usage patterns may vary throughout the lifecycle of the application.

Flexible Schema Design:

The NoSQL nature of DynamoDB allows for a flexible schema design, accommodating changes in user profiles and resume structures without requiring a major overhaul. This flexibility is crucial in the dynamic environment of a resume creator app where data models may evolve over time.

Attribute-based access control in DynamoDB enables fine-grained control over access to specific attributes, enhancing security while adapting to changing schema requirements.

Performance and Low-Latency Access:

DynamoDB is designed for low-latency access, ensuring a responsive user experience. This is critical in a resume creator app where quick access to user profiles and resume information is essential. The use of Global Secondary Indexes (GSIs) allows for efficient querying of data based on different attributes, contributing to optimized performance for various application use cases.

Managed Service Benefits:

As a managed service, DynamoDB manages administrative tasks such as hardware provisioning, setup, and configuration, reducing operational overhead. This allows the development team to focus more on building application features rather than managing database infrastructure.

Built-in Security Features:

DynamoDB integrates with AWS Identity and Access Management (IAM) for access control, ensuring that only authorized entities can interact with the database. Encryption at rest and in transit enhances data security, meeting the stringent security requirements of a resume creator application where user data confidentiality is paramount.

Security

In my project, the utilization of Amazon Virtual Private Cloud (VPC) and AWS Key Management Service (KMS) serves as a pivotal strategy for enhancing the overall security of our cloud infrastructure.

VPC (Amazon Virtual Private Cloud):

The incorporation of VPC is fundamental to securing our project's network environment. By defining a virtual network with VPC, we gain granular control over our cloud resources connectivity. VPC enables the creation of isolated subnets, route tables, and network gateways. This isolation provides a secure environment for our EC2 instances and other resources, preventing unauthorized access and potential security breaches. Additionally, VPC empowers us to configure security groups and network access control lists (NACLs), allowing

us to define fine-grained rules for inbound and outbound traffic. Through these network controls, VPC contributes significantly to mitigating the risks associated with unauthorized access and ensures a robust network architecture.

KMS (AWS Key Management Service):

The integration of AWS Key Management Service (KMS) plays a crucial role in managing encryption keys and bolstering the security of our sensitive data. KMS facilitates the encryption of data at rest and in transit, ensuring that our information remains confidential and protected from unauthorized access. By utilizing KMS, we can create and control encryption keys used to encrypt and decrypt our data, providing an additional layer of security. This is particularly essential for scenarios where we need to safeguard sensitive information, such as credentials or personally identifiable information (PII). KMS also seamlessly integrates with other AWS services, allowing us to encrypt data stored in Amazon S3, EBS volumes, and other supported services. Here are some additional advantages of using VPC and KMS in my project.

Amazon Virtual Private Cloud (VPC):

Granular Control and Isolation:

VPC allows for the creation of isolated subnets, providing granular control over the network architecture. This ensures that different components of the resume creator application are logically separated, preventing unauthorized access and potential security breaches.Isolation facilitates a secure environment for EC2 instances and other resources, enhancing the confidentiality and integrity of the application's data.

Fine-Grained Network Controls:

VPC enables the configuration of security groups and network access control lists (NACLs), allowing for the definition of fine-grained rules for inbound and outbound traffic. These controls are crucial for implementing the principle of least privilege, ensuring that only necessary and authorized communication occurs between different components of the application.

Scalability and Flexibility:

VPC is designed to scale with the growth of the application. It allows for the addition of new subnets, the modification of route tables, and the expansion of network resources to accommodate increasing demands. The flexibility of VPC architecture supports the evolving needs of the resume creator application as it scales in terms of users and features.

Security Best Practices:

VPC follows security best practices by default, providing features such as network isolation, security groups, and NACLs. This ensures a strong security foundation for the resume creator application. Integration with AWS Identity and Access Management (IAM) further enhances security by allowing fine-tuned control over user and resource permissions.

AWS Key Management Service (KMS):

Encryption for Sensitive Data:

KMS facilitates the encryption of data at rest and in transit, providing a robust mechanism to protect sensitive information stored in databases, file systems, or during data transmission. For a resume creator application, where personally identifiable information (PII) and sensitive credentials may be stored, the use of KMS ensures compliance with security and privacy regulations.

Control and Management of Encryption Keys:

KMS allows for the creation, rotation, and management of encryption keys used to safeguard data. This ensures that access to sensitive information is controlled and limited to authorized entities. The centralized management of keys simplifies the key lifecycle and enhances the overall security posture of the application.

Integration with AWS Services:

KMS seamlessly integrates with various AWS services, including Amazon S3, EBS volumes, and more. This allows for consistent encryption across different storage and compute resources, ensuring a comprehensive and standardized security approach.

Cost-Effective Encryption:

KMS operates on a pay-as-you-go model, making it a cost-effective solution. Users are billed based on the number of requests for key operations, ensuring that costs align with actual usage. The cost-effectiveness of KMS supports efficient budgeting for security measures within the resume creator application.

Networking & Content Delivery

For my project, utilizing Amazon VPC (Virtual Private Cloud) and Elastic Load Balancing (ELB) aligns with key architectural principles, ensuring a well-architected and efficient system.

Amazon VPC (Virtual Private Cloud):

Amazon VPC provides a dedicated and isolated section of the AWS Cloud for your project. This allows you to design a network environment with a high degree of control over IP addresses, subnets, and routing tables. In the case of my resume creation project, I can configure a VPC to securely host your application components. By defining subnets and employing security groups, you ensure that your resources are logically isolated, contributing to enhanced security. This is crucial for protecting sensitive user data and ensuring the confidentiality of resume-related information.

Elastic Load Balancing (ELB):

Elastic Load Balancing plays a pivotal role in distributing incoming application traffic across multiple targets, such as EC2 instances. In the context of my resume creation project, ELB ensures high availability and fault tolerance. By distributing traffic among different instances hosting your application, ELB optimizes resource utilization and provides a seamless experience for users. This is especially critical in scenarios where your application experiences varying levels of traffic – for instance, during peak usage times. ELB contributes to a responsive and reliable system architecture, ensuring that users can access and edit their resumes without disruptions. Here are some additional advantages of ELB.

Elastic Load Balancing (ELB):

High Availability and Fault Tolerance:

ELB distributes incoming traffic across multiple targets, such as EC2 instances, ensuring high availability and fault tolerance. In the event of a failure of one instance, traffic is automatically redirected to healthy instances, minimizing downtime and enhancing reliability.

Optimized Resource Utilization:

ELB optimizes resource utilization by evenly distributing traffic among different instances hosting the application. This ensures that all instances contribute to handling incoming requests, preventing overloading of any single instance and maximizing overall system efficiency.

Seamless User Experience:

By efficiently managing traffic distribution, ELB provides a seamless experience for users accessing and editing their resumes. Users can interact with the application without disruptions, even during peak usage times, ensuring a responsive and reliable system architecture.

Automatic Scaling:

ELB integrates seamlessly with Auto Scaling groups, allowing the automatic scaling of EC2 instances based on demand. This feature ensures that the application can handle varying levels of traffic effectively, providing scalability to accommodate increased user activity.

SSL/TLS Termination and Offloading:

ELB supports SSL/TLS termination, offloading the computational overhead of encrypting and decrypting traffic from the application servers. This enhances overall system performance and allows for the secure transmission of data between the user's device and the application.

Health Checks and Monitoring:

ELB performs health checks on instances to ensure they are responsive and healthy. Unhealthy instances are automatically removed from the load balancer's rotation, preventing them from receiving traffic until they are restored to a healthy state.

Integration with Amazon CloudWatch allows for monitoring and alerts based on various metrics, providing visibility into the health and performance of the load balancer and associated instances.

Security and SSL Certificates:

ELB supports the use of SSL certificates for encrypting data in transit. This is crucial for maintaining the security and confidentiality of sensitive information, such as login credentials and personal data, in the resume creator application. Security groups and NACLs can be configured to control inbound and outbound traffic, further enhancing the security posture of the application.

Cost-Effective Scaling:

ELB contributes to cost-effective scaling by automatically distributing incoming traffic to instances based on demand. This ensures that resources are efficiently utilized, preventing unnecessary costs associated with over-provisioning or under-provisioning of instances.

Architecture

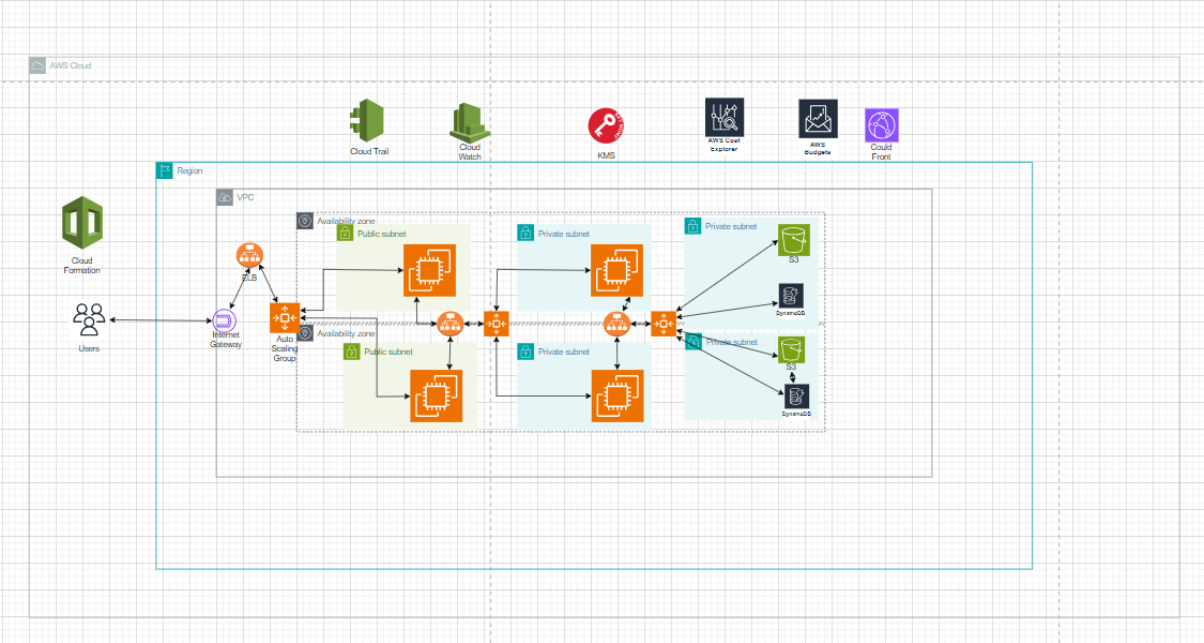


Figure 1 : Architecture Diagram

Workflow

Request Processing:

When users send requests to access the programme over the internet, the procedure starts. In front of the Internet Gateway, these requests pass via the CloudFront, which functions as a content delivery network (CDN). The Virtual Private Cloud (VPC) can be accessed through the Internet Gateway, which facilitates smooth communication in both directions between the VPC and the internet. This configuration makes use of CloudFront's distributed edge locations to improve overall performance and optimise content delivery.

Traffic Handling:

The Elastic Load Balancer (ELB) receives the traffic after it has entered the AWS environment. By allocating incoming application requests across several Host Servers in an Auto Scaling group on the Public Subnet, the ELB serves as a traffic cop. This guarantees that no server is overloaded and that traffic is diverted to the other servers that are still operational in the event of a server failure.

Front-end:

The application's front end, which may consist of web servers or application servers that produce dynamic content, is normally served by the host servers.

The Auto Scaling functionality automatically modifies the number of instances to guarantee constant, predictable performance in the event that a host server fails or reaches capacity.

By scaling the resources in or out, host server replicas in the Auto Scaling group can be utilised to boost fault tolerance and handle variable demands.

Back-end:

The application logic is handled by the Backend Servers in the Private Subnets, which handle the requests that are routed from the Host Servers. By preventing these vital parts from being immediately exposed to the public internet, private subnets improve security. An Auto Scaling group can additionally include the Backend Servers in order to control load and preserve performance. Backend Server Replicas give the application's back end more capacity and redundancy.

Database:

The data for the application is kept in the database. The Backend Servers use it to access data storage and retrieval functions. Since the database frequently holds sensitive data, it is situated in a different private subnet for security purposes.

For read-intensive applications, a database replica is utilised to spread read requests across the primary and replica databases, enhancing performance and offering a fallback option in the event that problems arise with the primary database.

Security and Encryption:

This architecture integrates AWS Key Management Service (KMS) to manage cryptographic keys that protect your data. KMS may communicate with services like the Database and Host Servers to encrypt and decrypt data as needed.

Monitoring, Management, and Compliance:

Every activity taken by users, roles, or AWS services is captured by AWS CloudTrail, which is an essential tool for auditing and improving security monitoring initiatives.

AWS CloudWatch is essential for tracking metrics and examining logs in order to monitor the performance of applications. It provides insightful information about the use of resources, the effectiveness of applications, and the general state of operations.

Code is used to automate infrastructure provisioning with AWS CloudFormation. It offers an efficient way to create and manage a set of resources by specifying every resource in a template. This makes it possible to provision and update resources in an orderly and predictable manner, which helps to ensure dependable and effective infrastructure management.

Cost Management:

To efficiently manage costs, AWS Cost Explorer is used to analyse and identify consumption trends and chances for cost savings.

You can create custom budgets with AWS Budgets and receive alerts when your expenditures or consumption surpass your allocated amount, or are expected to surpass it.

The application is guaranteed to be secure, scalable, and fault-tolerant by this architecture. High availability is provided by deployment in distinct Availability Zones, and auto scaling aids in dynamic load management. Services like KMS and CloudTrail are integrated across the architecture to provide security and compliance. Another essential component is cost management, which makes sure that the application's operating costs are constantly under control.

How the architecture is adhering the architecting principles and best practices.

The AWS Cloud architecture adheres to the AWS Well-Architected Framework's five pillars as follows:

1. Operational Excellence

The use of Auto Scaling Groups and Elastic Load Balancers (ELB) enables the system to adapt to varying load conditions, ensuring that the application is operating efficiently and maintaining performance. CloudWatch would be set up for monitoring the health and performance of the services, providing the necessary insights for operational decisions.

2. Security

Identity and Access Management (IAM):As I don’t have access to create any IAM roles, I was unable to implement it in the project. Although using IAM would secure the architecture very well.

Key Management Service (KMS): is used for creating and managing encryption keys, which is crucial for protecting data and ensuring compliance with industry standards. The architecture presumably uses security groups, NACLs, and other VPC features to secure the network layer.

3. Reliability

The architecture is designed across multiple Availability Zones to ensure high availability and fault tolerance. If one zone experiences an outage, the others can take over. Auto Scaling ensures that there is enough capacity to handle the workload even in the event of individual instance failures or spikes in demand.

4. Performance Efficiency

Resources are automatically scaled by the Auto Scaling Group based on demand, ensuring that performance is maintained without manual intervention. The use of Elastic Load Balancing efficiently distributes incoming network traffic across multiple targets, preventing any one instance from becoming a bottleneck.

5. Cost Optimization

AWS Cost Explorer and AWS Budgets provide insights into cost management, allowing for continuous cost optimization. They can be used to track usage and costs, and to set up alerts if spending exceeds thresholds.

The use of Auto Scaling also contributes to cost optimization by adding or removing resources to match demand, avoiding unnecessary costs associated with under or over-provisioning.

Here is how each component is contributing to follow the pillars of well architected framework

Elastic Load Balancer (ELB): Placed at the entry point of the architecture, the ELB distributes incoming application traffic across multiple targets, such as EC2 instances. This contributes to high availability and fault tolerance, key components of the Reliability pillar.

Auto Scaling Group: By automatically adjusting the number of EC2 instances, the Auto Scaling Group ensures that the application has the right amount of resources to handle the load at any given time. This supports both the Performance Efficiency and Cost Optimization pillars by scaling resources to match demand without over-provisioning.

EC2 Instances in Public and Private Subnets: EC2 instances host the application workloads. Instances in public subnets are directly accessible from the internet for tasks that require external access, while those in private subnets are isolated for internal processes. This setup enhances Security by reducing exposure to threats.

Virtual Private Cloud (VPC): The VPC provides a logically isolated section of the AWS Cloud where resources can be launched in a defined virtual network. This isolation is fundamental to the Security pillar.

Internet Gateway: This connects the VPC to the internet, enabling EC2 instances in public subnets to send and receive traffic. The gateway is essential for the Operational Excellence pillar, as it allows for management and external monitoring of resources within the VPC.

Simple Storage Service (S3) and DynamoDB: These services provide scalable storage solutions, with S3 for object storage and DynamoDB for NoSQL database services. S3 and DynamoDB adhere to the Performance Efficiency pillar by offering high-performance storage solutions that scale seamlessly.

CloudWatch: This monitoring service provides data and actionable insights to monitor applications, respond to system-wide performance changes, and optimize resource utilization. CloudWatch is integral to the Operational Excellence pillar.

Key Management Service (KMS): This service helps manage cryptographic keys for your applications, which is a critical aspect of the Security pillar.

AWS Cost Explorer and AWS Budgets: These services provide insights into your AWS cost and usage, and allow you to set custom budgets, contributing to the Cost Optimization pillar by helping manage and control costs.

Availability Zones: By distributing EC2 instances across multiple Availability Zones, the architecture ensures that the application can withstand the failure of a single zone, which is crucial for the Reliability pillar.

References:

[1] "Getting More Out of Amazon EC2 (30:59)," Amazon Web Services, Inc., 2023. https://aws.amazon.com/pm/ec2/?gclid=CjwKCAiAjrarBhAWEiwA2qWdCHkKiyqnruBDCyZBT8ZWnSjtAbHckK4GKMVx-ePBvFV7bQQaiig3rRoCotIQAvD_BwE&trk=8c0f4d22-7932-45ae-9a50-7ec3d0775c47&sc_channel=ps&ef_id=CjwKCAiAjrarBhAWEiwA2qWdCHkKiyqnruBDCyZBT8ZWnSjtAbHckK4GKMVx-ePBvFV7bQQaiig3rRoCotIQAvD_BwE:G:s&s_kwcid=AL!4422!3!536323340610!p!!g!!amazon%20ec2%20vps!11346198414!112250792158 (accessed Dec. 01, 2023).

[2] free, "Flowchart Maker & Online Diagram Software," Diagrams.net, 2023. https://app.diagrams.net/ (accessed Nov. 30, 2023)

[3] "Accessing DynamoDB - Amazon DynamoDB," Amazon.com, 2023. https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/AccessingDynamoDB .html (accessed Nov. 29, 2023).

[4] "Protecting data with server-side encryption - Amazon Simple Storage Service," *Amazon.com*, 2023. https://docs.aws.amazon.com/AmazonS3/latest/userguide/serv-side-encryption.html (accessed Dec. 04, 2023).

[5] "Security - AWS Well-Architected Framework," *Amazon.com*, 2020. https://wa.aws.amazon.com/wellarchitected/2020-07-02T19-33-23/wat.pillar.security.en.html (accessed Dec. 03, 2023).