Cloud Solution:

Creating an AWS Architecture for a global startup medical company

ITC 6480 - Professor Taglienti
Alejandro Gerov - Applied Machine Intelligence
Shivam Sinha - Informatics (specialization in Cloud computing)

Table of Contents

Introduction

The purpose of this project is to design and implement a secure, scalable, and highly available cloud infrastructure using Amazon Web Services (AWS) to support a modern web application. This report details the architecture, configuration, and security measures implemented to meet the specific requirements of the project, ensuring a robust environment for both production and development operations.

The project leverages key AWS services, including Virtual Private Cloud (VPC), Application Load Balancers (ALBs), Amazon RDS, IAM roles, and Amazon Route 53 to create a well-organized infrastructure that provides the necessary isolation between environments while maintaining high availability and performance. Route 53 plays a crucial role in managing DNS and directing traffic efficiently across multiple AWS regions—North America (NA), Europe, Middle East, and Africa (EMEA), and Asia-Pacific (APAC)—ensuring low latency and high availability for a global user base. The architecture also integrates security best practices, such as the use of private subnets and carefully configured security groups, to protect sensitive data and application resources. The design and implementation align with the AWS Well-Architected Framework, specifically focusing on the pillars of Security, Reliability, Performance Efficiency, and Cost Optimization. This ensures that the infrastructure not only meets the immediate needs of the application but is also resilient, efficient, and cost-effective over time.

This report outlines the steps taken to configure the network, compute, and storage resources, and explains how these components work together to deliver a reliable and secure application environment. The project also addresses business continuity, with a focus on fault tolerance and disaster recovery, ensuring that the infrastructure can withstand potential failures without significant impact on application availability.

Overall, the project demonstrates a comprehensive approach to cloud architecture design, reflecting an understanding of both the technical and operational requirements needed to successfully deploy and manage a complex web application in the cloud.

Executive Summary

The configuration of the VPCs, subnets, instances, and security settings reflects a well-structured and secure architecture designed to meet the specific needs of the project. The choice to use two VPCs—one for production and one for development—ensures a clear separation of environments, which enhances security and enables more efficient resource management. Both VPCs are located in the us-east-1 region, which is known for its availability and proximity to a significant user base, ensuring low latency.

Each VPC is segmented into four subnets, spread across two Availability Zones (AZs) to provide high availability and fault tolerance. The subnets are further divided into public and private categories, where the public subnets are used for resources that need to be accessible from the internet (such as the web servers and load balancers), and the private subnets are reserved for more sensitive resources (like the application servers and databases). This segmentation helps isolate and protect the critical components of the application from direct exposure to the internet.

The web and application instance configurations were carefully selected to balance performance and cost-efficiency. The web tier uses t3.medium instances, which provide sufficient resources (2 vCPU, 4 GB RAM) to handle web traffic efficiently. The application tier uses slightly more powerful t3.large instances (2 vCPU, 8 GB RAM) to manage the increased processing load of handling application logic and API requests. The database tier is managed through Amazon RDS with a db.t3.large instance, configured in a Multi-AZ setup to ensure high availability and disaster recovery. This approach aligns with best practices for database management in the cloud, where reliability is paramount.

The load balancer configurations ensure that traffic is distributed evenly across instances in the web and application tiers. The web tier load balancer is external, meaning it handles internet-facing traffic, while the application tier load balancer is internal, used to distribute traffic between the application instances in the private subnets. This setup not only enhances availability but also improves security by keeping the application tier isolated from direct internet access.

Security group configurations are also carefully crafted to minimize exposure while allowing necessary communication between tiers. The web tier security group allows HTTPS traffic from any source, reflecting its public-facing nature. The application tier security group restricts traffic to HTTPS only from the web tier, ensuring that only web servers can communicate with the application servers. Similarly, the database tier security group allows MySQL traffic only from the application tier, further securing the database from unauthorized access.

Finally, business continuity is ensured through the use of appropriately sized instances and a robust security configuration. The use of t3.medium instances for the web tier and t3.large for the app tier ensures that the system can handle peak loads while maintaining cost efficiency. Security groups are aligned with each tier's role, ensuring that each component of the architecture is protected from unauthorized access.

Overall, this architecture is designed to be secure, scalable, and resilient, meeting the requirements set forth by the project while adhering to AWS best practices.

Overall Requirements & Assumptions

The project aimed to establish a secure, scalable, and compliant AWS infrastructure for a Medical startup company, a SaaS provider offering a medical social networking and diagnosis assistance application across APAC, the US, and Europe. The application connects patients and doctors for online appointments, remote consultations, diagnoses, electronic prescription transfers, and payment services. With sensitive medical data involved and an anticipated user growth post-launch, the infrastructure needed to be robust and secure.

The AWS architecture was designed to mirror the current on-premises three-tier setup (Web, Application, Database tiers) using Microsoft Windows servers with IIS and SQL Server Standard Edition. The new environment includes separate VPCs for development, testing, and production, spread across multiple Availability Zones to ensure high availability.

User groups, including System Administrators, Database Administrators, and a Monitoring Group, were given distinct access levels, with secure authentication enforced via MFA and strict password policies. Security was paramount, with private subnets, security groups, and load balancers supporting HTTP, HTTPS, and TCP protocols to manage traffic efficiently.

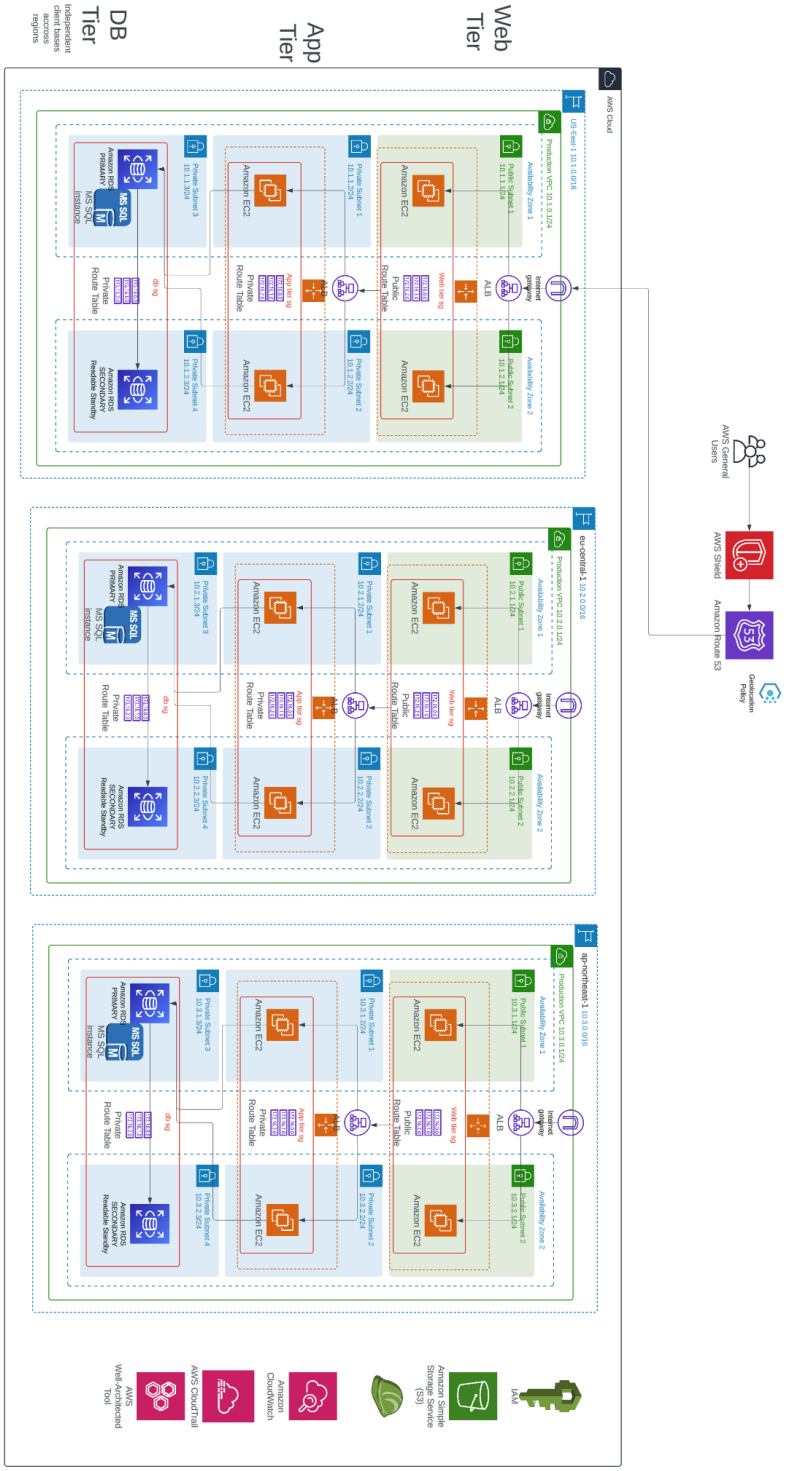
The infrastructure incorporated Multi-AZ configurations for Amazon RDS to ensure database resilience and business continuity. Continuous auditing and monitoring were implemented using AWS CloudWatch and CloudTrail to track activities and securely store logs. Integration with Amazon S3 was also established for efficient document and image processing.

While the current architecture is regionally segregated, the design considers future scalability needs, allowing for potential inter-regional operations as the company grows.

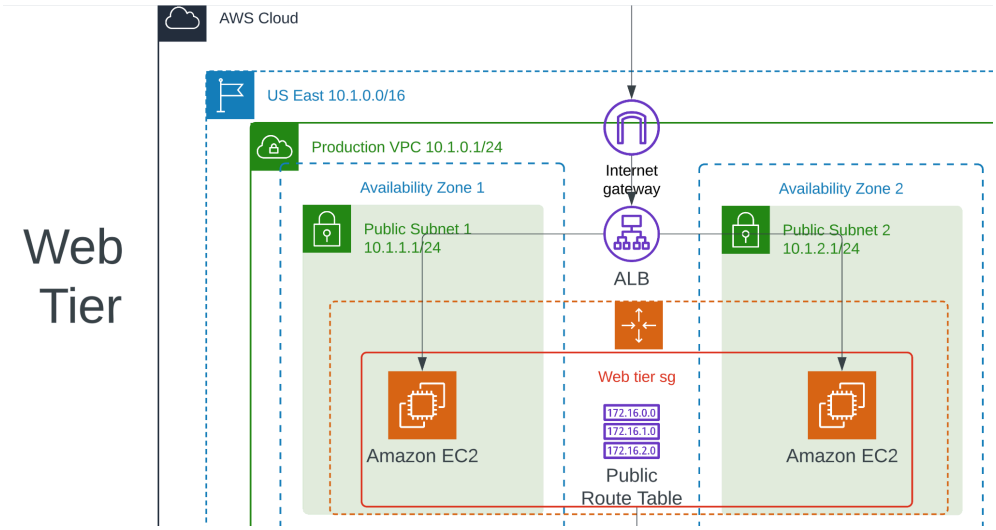
Overall, this project delivers a secure, scalable, and compliant cloud architecture, ready to support the application’s launch and future expansion.

Solutions - [Link to interactive diagram]

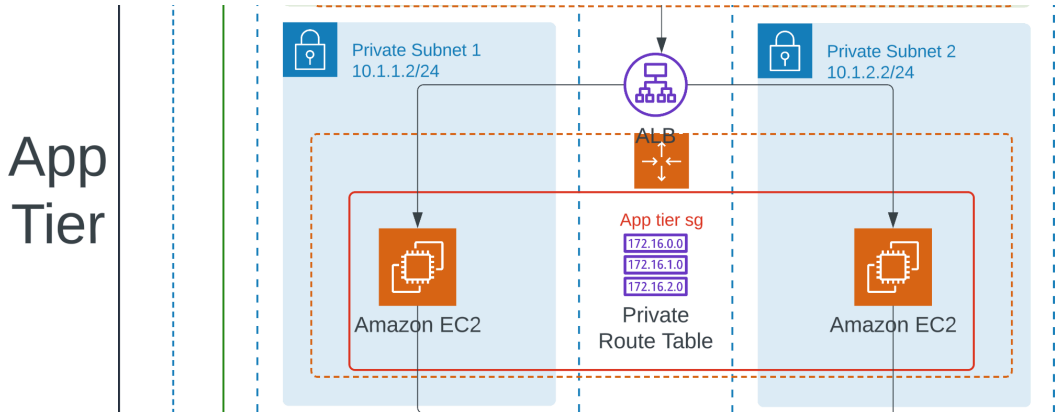
Complete Three Tier for production multi region architecture



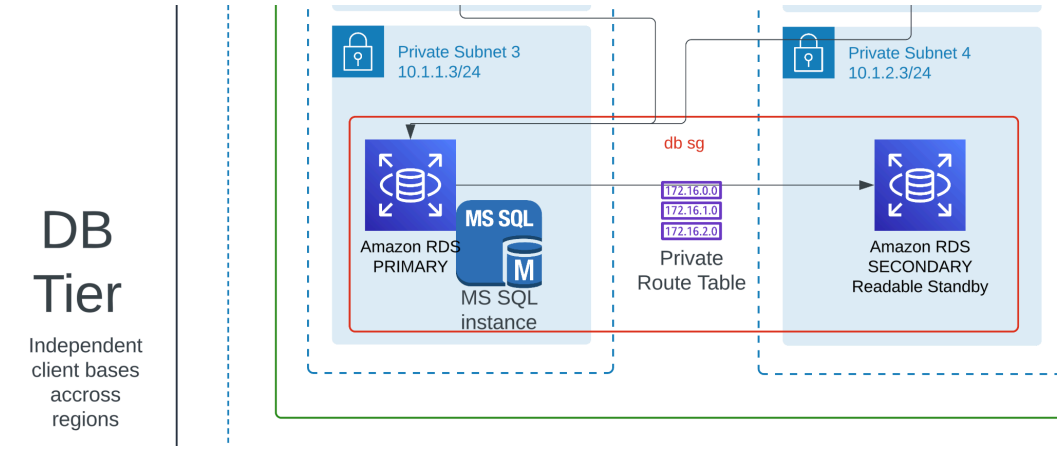
Web Tier - Zoom of us-east region



App Tier - Zoom of us-east region



Database Tier - Zoom of us-east region



AWS Services

Given that our startup is expecting an international crowd with anticipation coming from distinct global regions, using Route 53 as a DNS to geographically route users to the nearest region (The company is expecting customers in NA, APAC & EMEA). This will be along an AWS Shield to prevent denial-of-service attacks. The network will then route through a load balancer within the specific region of the web tier. The web will communicate with the app tier subsequently, while also using a load balancer within the region to maintain high availability of services. Web and App tiers will have EC2 instances within each availability zone, while having a regional autoscaling group and a security group. The application tier will communicate to the primary RDS instance of Microsoft SQL, and in case of some downtime, it will failover to the secondary which is in readable standby. As mentioned previously, one of our assumptions is that the clients will remain geo-static (They will not move: patients will not seek healthcare outside of professionals within their region.)

- AWS Shield
- Route 53
- IAM
- S3
- Amazon Cloudwatch
- AWS Cloudtrail
- AWS Well Architected Tool
- (AWS Configs)

Web Tier/App Tier

- Virtual Private Cloud
- Auto scaling groups
- Amazon EC2
- Application Load Balancer
- Internet Gateway (Web tier only)
- Route Table (Public for Web, Private for App)

Database Tier

- Amazon RDS for Microsoft SQL
- Security group

User Authentication

The User Authentication section of this project outlines the comprehensive security measures implemented to manage and control access to AWS resources. This approach is designed to ensure that only authorized users can access the system, with appropriate permissions tailored to their roles. The project defines three key user groups—System Administrators, Database Administrators, and the Monitoring Group—each with specific permissions and access levels aligned with their responsibilities.

For enhanced security, multi-factor authentication (MFA) is enforced for all administrators to add an extra layer of protection beyond traditional passwords. Additionally, IAM roles, such as webAccessRole, appAccessRole, and dbAccessRole, have been created to allow applications to access AWS resources securely, without embedding authentication credentials directly into the application code.

The IAM password policy is strictly enforced to meet security best practices, requiring a combination of upper and lowercase letters, numbers, and special characters, with mandatory password changes every 90 days. These measures ensure that user authentication and access control are robust, reducing the risk of unauthorized access and protecting the integrity of the AWS environment.

Roles to be created

G: sysadmins	G: db-admins	G: monitoring	R: Roles for Resource Access
admin1,admin2	db-manager1, db-manager2	auditor1, auditor2, auditor3, auditor4	webAccessRole appAccessRole dbAccessRole

Group/Role #	Group/Role Name	Permissions
Group	sysadmins	Full administrative access to AWS services. **Policy**: Attach the **AdministratorAccess** managed policy to this group. **MFA Requirement**: Enforce Multi-Factor Authentication (MFA) using a Virtual MFA device.
Group	db-admins	Full access to Amazon RDS, and limited access to EC2 (for managing database instances only). **AmazonRDSFullAccess**

		EC2 instance management permissions (ec2:DescribeInstances, ec2:CreateSecurityGroup, etc.) **CloudWatch Logs Access**
Group	monitoring	Read-only access to monitor AWS infrastructure, including EC2, RDS, S3, CloudWatch, and VPC Flow Logs. **ReadOnlyAccess** **MFA required**
Role	webAccessRole appAccessRole dbAccessRole	For Application to access resources without embedding auth

Requirement	Solution
Should be at least 8 characters and have 1 uppercase, 1 lowercase, 1 special character, and a number.	IAM password Policy: Minimum password length Character requirements (click checkbox, and adjust integer input)
Change passwords every 90 days and ensure that the previous three passwords can't be re-used.	IAM password Policy "Enforce password expiration" - adjust integer input to 90 days Prevent password reuse - click checkbox
All administrators require programmatic access	**AdministratorAccess** policy for System administrator When setting up admin users, check box for "Programmatic access"
Administrator sign-in to the AWS Management Console requires the use of Virtual MFA.	Request administrators to set up MFA on their account on first login. Enforce by creating IAM policy which denies if MultiFactorAuthPresent: 'false'

Network and Security

The network architecture for this project is meticulously designed to ensure optimal performance, security, and scalability across different regions, tailored to the specific needs of the organization's global operations. The architecture is distributed across three primary regions: North America (Virginia), EMEA (Frankfurt), and APAC (Tokyo), with each region hosting its own Production VPC. An additional Testing VPC is located in Ohio, mirroring the production environments to ensure consistency in testing and development.

Each Production VPC is segmented into six subnets, evenly distributed across two Availability Zones (AZs) to enhance redundancy and availability. The subnet allocation follows a consistent pattern across regions, with specific CIDR ranges assigned to each subnet. For example, the Virginia Production VPC uses the CIDR range `10.1.0.0/24`, while Frankfurt and Tokyo use `10.2.0.0/24` and `10.3.0.0/24`, respectively.

Within each VPC, subnets are categorized based on their roles in the web, application, and database tiers. Public subnets are designated for web servers and allow external access, while private subnets host application servers and databases, ensuring that these critical resources are not directly exposed to the internet. The subnet names are unique across regions but follow a standardized naming convention within each region for easy identification and management.

For the Testing VPC in Ohio, the subnet structure replicates that of the production environments, maintaining the same public and private subnet distribution across AZs. This consistent configuration across all regions and environments ensures that the network architecture is both robust and easy to manage, while also being flexible enough to accommodate future expansion and changes in the organization's needs.

VPC	Region (MS SQL Server compatible)	Purpose	Subnets	AZs	CIDR Range
1	us-east-1 Virginia	Production VPC for NA	6	2	10.1.0.1/24
2	eu-central-1 Frankfurt	Production VPC for EMEA	6	2	10.2.0.1/24
3	ap-northeast-1 Tokyo	Production VPC for APAC	6	2	10.3.0.1/24
0 (not shown in architecture	us-east-2 Ohio	Testing VPC (identical to other regional VPCs)	6	2	10.0.0.0/24

Purpose Tier	Subnet Name Unique across regions Same within region	VPC	Subnet Type (Public/private)	AZ	Subnet Address
Web prod	public-subnet-1	1,2,3 (repeated across regional VPCs)	public	1	10.#.1.1/24 0=testing 1=prod NA 2=prod EMEA 3=prod
Web prod	public-subnet-2	1,2,3	public	2	10.#.2.1/24
App prod	private-subnet-1	1,2,3	private	1	10.#.1.2/24
App prod	private-subnet-2	1,2,3	private	2	10.#.2.2/24
Db prod	private-subnet-3	1,2,3	private	1	10.#.1.3/24
Db prod	private-subnet-4	1,2,3	private	2	10.#.2.3/24

Purpose Tier	Subnet Name	VPC	Subnet Type (Public/private)	AZ	Subnet Address
web	public-subnet-1	0 (testing)	public	1	10.0.1.1/24
web	public-subnet-2	0 (testing)	public	2	10.0.2.1/24
app	private-subnet-1	0 (testing)	private	1	10.0.1.2/24
app	private-subnet-2	0 (testing)	private	2	10.0.2.2/24
db	private-subnet-3	0 (testing)	private	1	10.0.1.3/24
db	private-subnet-4	0 (testing)	private	2	10.0.2.3/24

Web and Application

The web and application layers of this architecture are meticulously designed to deliver robust, scalable, and secure services across different environments. Each tier—web, application, and database—has been configured with instances and security measures that align with the organization's performance and security requirements.

Web Tier: The web tier comprises two instances running Amazon Linux 2 on t3.medium instances, each with 2 vCPUs and 4 GB of RAM. This setup is optimized for managing web traffic effectively. Both instances are part of an autoscaling group, enabling dynamic adjustments based on demand, ensuring consistent availability and performance. Traffic is distributed via an external Application Load Balancer (web-alb), configured to allow HTTP (port 80) and HTTPS (port 443) traffic, routed through Route 53 or the Internet Gateway, enhancing fault tolerance.

Application Tier: The application tier also includes two instances, each running Amazon Linux 2 on t3.large instances with 2 vCPUs and 8 GB of RAM. These instances are designed to handle the processing of application logic and API requests efficiently. Similar to the web tier, the application tier uses an autoscaling group to maintain performance under varying loads. An internal Application Load Balancer (app-alb) manages traffic within private subnets, ensuring that only traffic from the web tier can access the application tier, adding a crucial layer of security.

Database Tier: The database tier is powered by an Amazon RDS instance running Microsoft SQL on a db.t3.large instance type with 2 vCPUs and 8 GB of RAM. Configured in a Multi-AZ deployment, this setup ensures high availability and fault tolerance. The database security group (db-sg) strictly allows traffic on port 1433 (MS SQL Server) from the application tier, ensuring that the database remains isolated and secure.

This architecture balances performance, security, and scalability, with built-in redundancy and autoscaling to handle fluctuations in demand while maintaining optimal performance.

Instances for each tier

Tier	Tag*	OS	Type	Size	Justification	# of instances	User Data?
Web	Key = Name Value = web-tier	Amazon Linux 2	t3.medium	2 vCPU, 4 GB RAM	Suitable for handling web traffic	2 within autoscaling group	No

App	Key = Name Value = app-tier	Amazon Linux 2	t3.large	2 vCPU, 8 GB RAM	Handles application logic and API requests	2 within autoscaling group	No
DB	Key = Name Value = db-tier	Amazon RDS (for Microsoft SQL)	db.t3.large	2 vCPU, 8 GB RAM	Managed service, Multi-AZ for high availability	1 (Multi-AZ)	N/A

Load Balancer	Name*	Ext/Int	Subnets	SG Name*	Rule	Source
Web Tier	web-alb	External	public-subnet-1 public-subnet-2	web-tier-sg	Allow HTTP (80) Allow HTTPS (443) Allow TCP(4172)	Route 53 / Internet Gateway 0.0.0.0/0
App Tier	app-alb	Internal	private-subnet-1 private-subnet-2	app-tier-sg	Allow HTTP (80) Allow HTTPS (443) Allow TCP(4172)	web-tier-sg

Instance Tier	SG Name*	Rules	Source
Web Tier	web-tier-sg	Allow HTTP (80) Allow HTTPS (443)	Route 53 / Internet Gateway 0.0.0.0/0
App Tier	app-tier-sg	Allow HTTP (80) Allow HTTPS (443)	web-tier-sg
DB Tier	db-sg	Allow HTTP (80) Allow HTTPS (443) MS SQL Server(1433)	app-tier-sg

Business Continuity

In ensuring business continuity, auto-scaling plays a critical role by maintaining application availability and performance under varying loads. The architecture's auto-scaling groups are configured to automatically adjust the number of running instances in response to demand, minimizing the risk of service disruptions and ensuring seamless scalability.

Web Tier:

- **OS and Configuration:** The web tier operates on Amazon Linux 2, using t3.medium instances (2 vCPU, 4 GB RAM). These instances are part of the "WebTier" configuration and are assigned the "webAccessRole" to facilitate secure and efficient resource access.
- **Auto-Scaling Group:** The "WebTier ASG" manages the web tier's scaling, with a minimum of 2 instances and a maximum of 4. The instances are distributed across two public subnets (public-subnet-1 and public-subnet-2) within the VPC. An Application Load Balancer (ALB) ensures that incoming traffic is balanced across all active instances, improving fault tolerance. Tags are used for identification, with the web tier associated with the "Web Tier ELB" for load balancing.

Application Tier:

- **OS and Configuration:** The application tier also runs on Amazon Linux 2, utilizing t3.large instances (2 vCPU, 8 GB RAM). These instances are part of the "AppTier" configuration and use the "appAccessRole" to manage secure access to necessary resources.
- **Auto-Scaling Group:** The "AppTier ASG" oversees the scaling of the application tier, with the same instance configuration as the web tier—minimum 2 and maximum 4 instances. The instances are placed within two private subnets (private-subnet-1 and private-subnet-2) in the VPC, with an internal Application Load Balancer (app-tier alb) handling traffic distribution. Tags are applied to identify the production environment.

By leveraging these auto-scaling groups, the architecture ensures continuous operation even during periods of high demand, with the flexibility to scale resources as needed while maintaining security and performance standards. This setup is essential for achieving business continuity and resilience in dynamic cloud environments.

Tier	OS	Type	Size	Configuration Name*	Role	Security Group
Web	Amazon Linux 2	t3.medium	2 vCPU, 4 GB	WebTier	webAccessRole	web-tier-sg
App	Amazon Linux 2	t3.large	2 vCPU, 8 GB	AppTier	appAccessRole	app-tier-sg

Tier	LaunchConfiguration*	Group Name*	Group Size	VPC	Subnets	ALB	Tags
Web	Web	WebTier ASG	Min 2 Max 4	2	Public-subnet-1 public-subnet-2	Web-tier alb	Web Tier ELB
App	App	AppTier ASG	Min 2 Max 4	2	private-subnet-1 private-subnet-2	App-tier alb	Production

Auditing

For auditing, we have set up a robust system to ensure continuous monitoring and logging of all account activities across our AWS infrastructure. The specific auditing requirements are as follows:

1. Continuous Monitoring: We utilize AWS CloudWatch and CloudTrail to continuously monitor account activities. CloudWatch provides real-time monitoring, while CloudTrail records all actions taken within the AWS environment.
2. Event History Logging: AWS CloudTrail is configured to log the event history of AWS account activities. This includes actions taken through the AWS Management Console, AWS SDKs, command line tools, and other AWS services, ensuring a comprehensive record of all operations.
3. API Call Audit Trail: CloudTrail ensures that every executed API call is captured, providing a detailed audit trail that includes information on who performed an action, when it was done, and what resources were affected.
4. Secure Log Storage: All logs generated by CloudTrail are securely stored in encrypted Amazon S3 buckets, ensuring that the audit data is both protected and available for compliance and auditing purposes.

By implementing these measures, we ensure that administrators like admin1, admin2, database managers like db-manager1, db-manager2, and auditors like auditor1, auditor2, auditor3, auditor4 have a clear, secure, and auditable trail of all activities within the AWS environment.

Future Steps

Moving forward, we will focus on several key initiatives to enhance our AWS infrastructure:

1. Enhanced Monitoring Protocols: We intend to bolster our monitoring capabilities by implementing AWS Lambda for automated analysis and real-time notifications of suspicious activities. This proactive approach will help us swiftly address potential security threats, maintaining the integrity and reliability of our services.
2. Cost Analysis and Resource Utilization: We will prioritize a comprehensive review of our resource utilization to identify areas where we can optimize costs and improve efficiency. This will ensure that we are making the most of our resources without unnecessary expenditure.
3. Quality of Life Monitoring for Users and Customers: Leveraging the AWS Well-Architected Tool, we will closely monitor customer experience metrics, such as latency and performance across different geolocations. By pinpointing where users encounter the most issues, we can implement targeted improvements to enhance their overall experience.
4. Inter-Regional Scalability Considerations: While our current design assumes that inter-regional operations, like medical practices, are not yet required, we recognize the possibility of future needs in this area. We can plan for scalability and explore strategies to support inter-regional operations as they become necessary, ensuring that our RDS instances are prepared to meet evolving demands.

These steps will ensure that our infrastructure remains robust, cost-effective, and adaptable, while continuing to deliver a high-quality experience for all users and customers.