Ministry of Communications and Information Technology

aws academy

DEPI-Project

AWS

Building a Highly

Available, Scalable Web Application

BY:Hesham Abdalhmid Abdalgalil

Contents

Overview:

This archive diagrams the design and components utilized to construct a profoundly accessible, adaptable web application on AWS.

The extend points to make strides the execution of a understudy records web application amid top affirmations periods by leveraging AWS administrations.

Objective:

• Create an structural chart: Outline the intelligent between AWS services.

• Estimate costs: Utilize the AWS Estimating Calculator for taken a toll estimation.

• Deploy a useful web application: Have on a virtual machine sponsored by a social database.

• Ensure tall accessibility and adaptability: Execute stack adjusting and auto-scaling.

• Secure the application: Design arrange security and oversee get to consents.

Design planning and cost estimation:

Design planning A variety of tools are available for creating architectural designs. You would want to get the official AWS architectural symbol, locate references, and use a clear chart as one of the suggested tools. diagrams of architecture. can use the AWS Pricing Calculator to get the approximate cost in AWS.

Functional: The solution satisfies the functional specifications, including the capacity to see, add, remove, or

alter the student records as quickly as possible.

Load balanced: To prevent overload or underutilization, the solution may appropriately balance user traffic.

resources.

Scalable: The solution is made to grow with the application to accommodate its demands.

Highly available: The system is built to have little downtime in the event that a web server fails.

not available.

Secure: The database is protected and not immediately accessible from open networks. The web servers

and database are only accessible over the proper ports. The online application may be accessed via the

online. The web application does not have the database credentials hardcoded into it.

Cost-optimized: The solution is made to be as affordable as possible.

High performance: All standard functions (such as viewing, adding, removing, or changing records) are carried out

without noticeable lag under typical, fluctuating, and high loads

Presumptions:

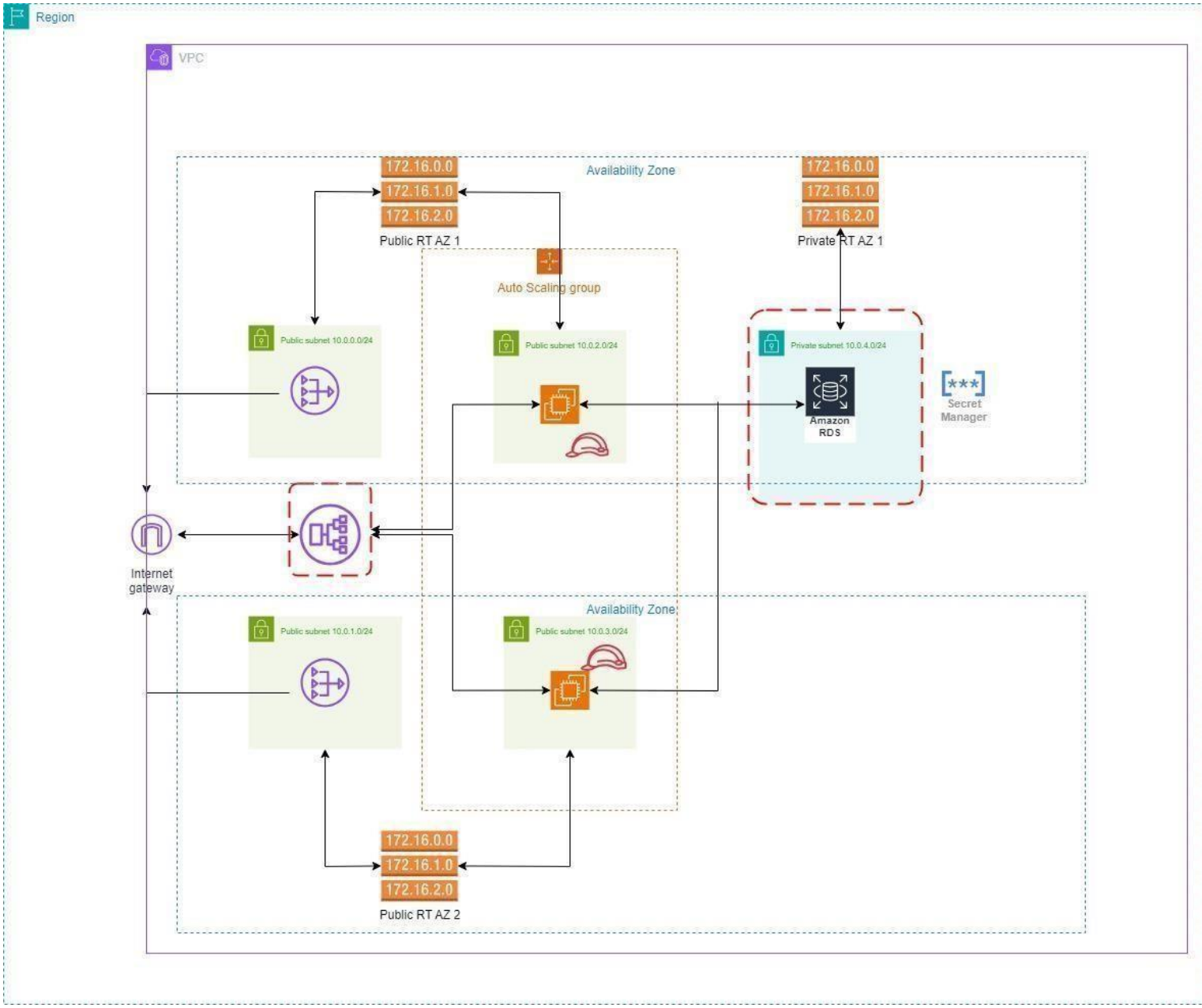
This project will be developed in a controlled laboratory setting with limitations on features, services, and financial resources.

Take into account the following project assumptions:

- One AWS Region is where the application is installed.
- Neither a bespoke domain nor HTTPS access are required for the website.
- The given JavaScript code is used to install the solution on Ubuntu computers.
- Unless otherwise specified in the instructions, use the JavaScript code exactly as stated.
- The solution makes advantage of features and services while staying within the confines of the laboratory.
- The website is open to the public without authentication; the database is only housed in one Availability Zone.

Architecture:

This area gives an diagram of the key AWS components utilized to construct a exceedingly accessible, versatile, and secure web application. Each component plays a vital part in guaranteeing the application meets execution and security prerequisites, particularly amid crest utilization periods. The design is planned taking after AWS best hones to optimize taken a toll, versatility, and tall availability.



- Virtual Private Cloud (VPC): Sets up a secure arrange environment with separated assets, counting open and private subnets over numerous Accessibility Zones for redundancy.

- Amazon EC2 Occasions: Has web applications, giving compute control with flexibility.

- Amazon RDS: Oversees the social database autonomously from the application servers. Arranged with MySQL motor in a multi-AZ sending for tall availability.

- AWS Insider facts Chief: Secures delicate data like database qualifications, empowering secure get to by the web application without hardcoding credentials.

- Application Stack Balancer (ALB): Disseminates approaching activity over numerous EC2 occurrences to guarantee stack adjusting and tall availability.

- Auto Scaling Gather: Consequently alters the number of EC2 occurrences based on request to keep up performance.

- AWS Cloud9: Gives a cloud-based improvement environment for overseeing and conveying code.

- AWS WAF (Web Application Firewall): Secures the web application from common web abuses and bot activity by sifting hurtful demands some time recently they reach the application.

- Amazon CloudWatch: Screens application execution and asset utilization, giving experiences and alarms to keep up operational wellbeing and efficiency.
- AWS Reinforcement: Centralizes reinforcement administration for AWS assets, guaranteeing information assurance and compliance through robotized planning of reinforcements for RDS and EC2 volumes.

Each component has been carefully chosen and designed to guarantee that the application remains responsive and secure, indeed beneath overwhelming stack. The taking after segments will dig into the reason and setup of each component in detail.

1. VPC

We chosen the Virtual Private Cloud (VPC) to disconnect the application inside a secure arrange environment. This guarantees that the web application is ensured from unauthorized get to and can as it were be come to through controlled section focuses.

Components

- **Public Subnets:** These are utilized to have assets that require to be open from the web, such as stack balancers or bastion has. Open subnets are designed with a course to an web door, permitting inbound and outbound activity.
- **Private Subnets:** These have assets that do not require coordinate get to from the web, such as databases and application servers. Private subnets are arranged without coordinate web get to, improving security by constraining presentation.

Availability Zones

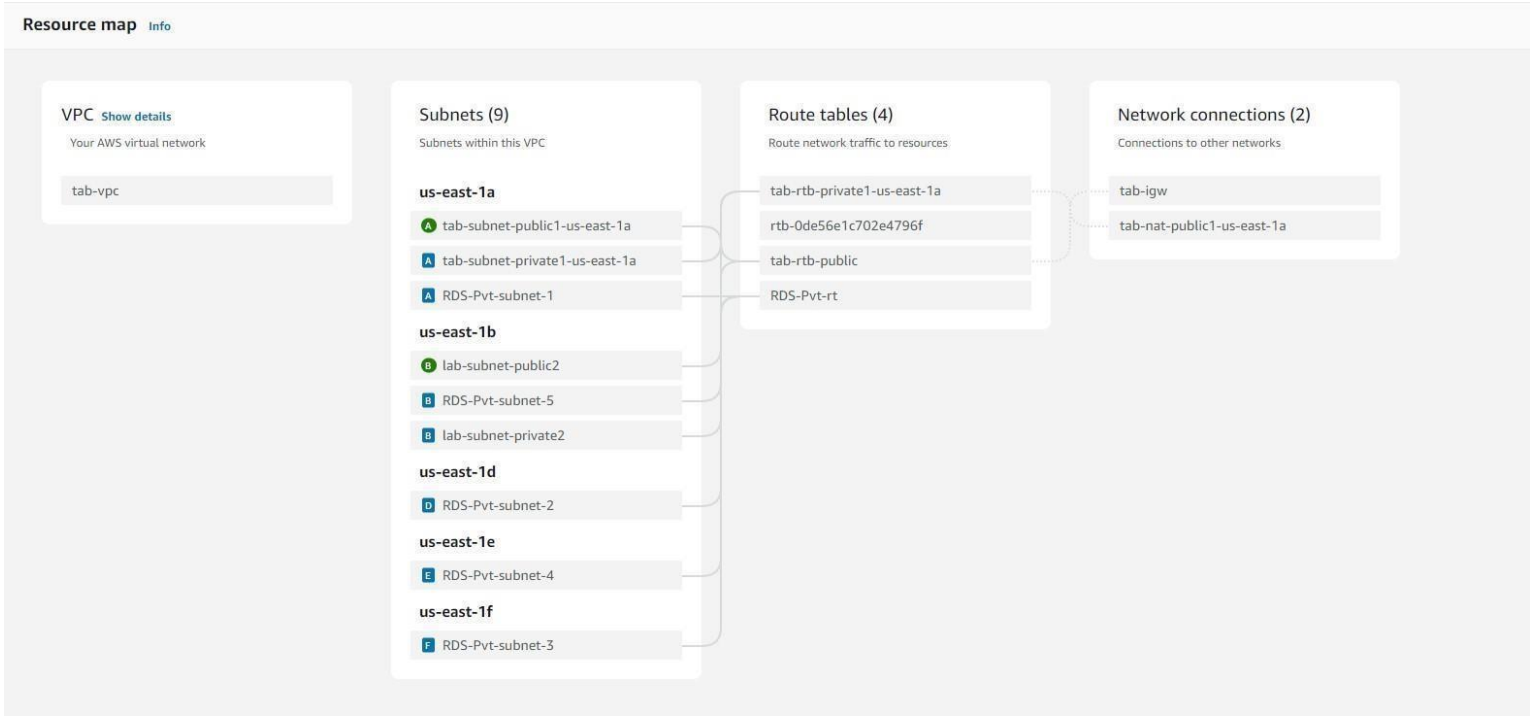
Conveying subnets over different Accessibility Zones (AZs) guarantees repetition and tall accessibility. If one AZ encounters an blackout, the assets in other AZs can proceed to work, minimizing downtime.

Benefits

- **Isolation:** The VPC provides a logically isolated network environment for the web application.
- **Security:** By using private subnets and security groups, sensitive components like databases are protected from direct internet access.
- **Scalability:** The VPC can be easily scaled by adding more subnets or modifying existing configurations to accommodate growth.

Conclusion

The utilize of a VPC with both open and private subnets over different Accessibility Zones gives a strong establishment for building secure, versatile, and profoundly accessible applications on AWS. This setup adjusts with the best hones laid out in the AWS Well-Architected System, guaranteeing ideal execution and security for the understudy records web application.



2. EC2:

We utilized Amazon EC2 occurrences to have the web server and application code, giving the vital compute control to run the application proficiently. These occurrences guarantee that the application can handle changing loads by scaling up or down as required.

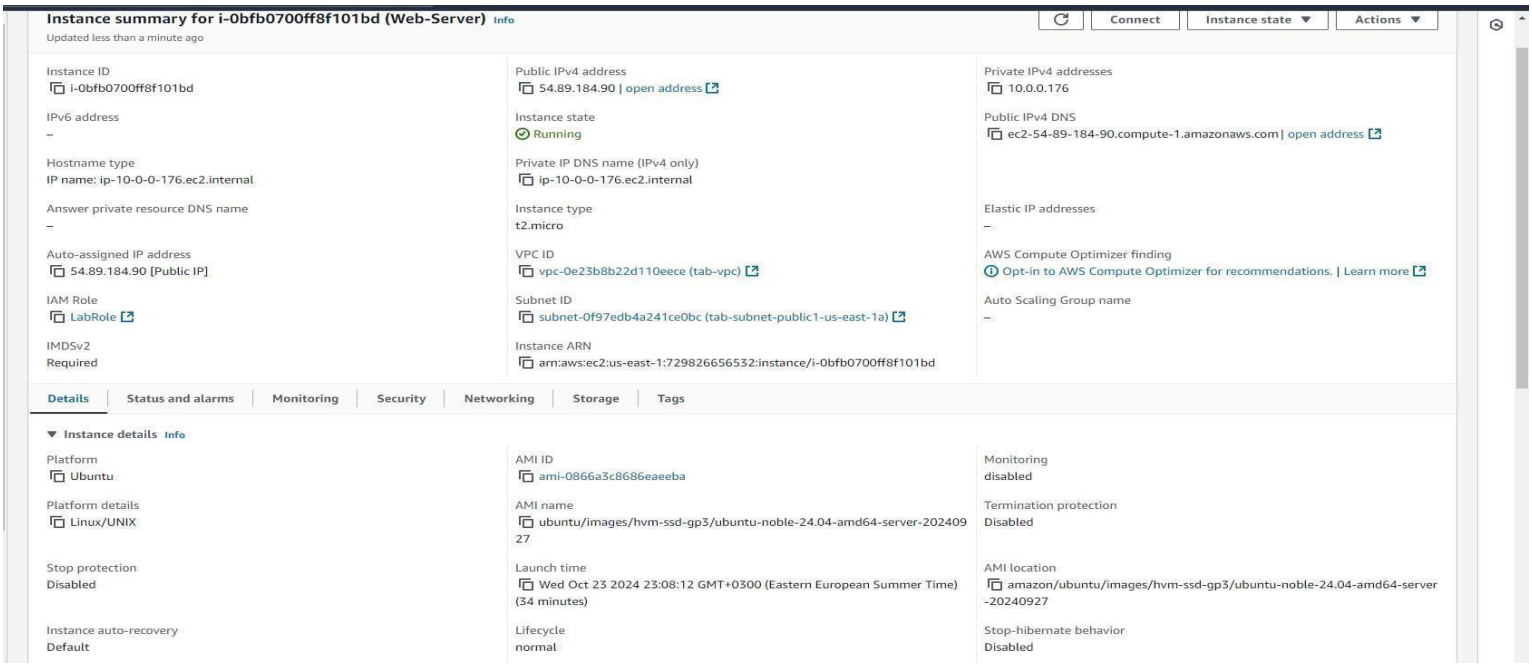
Security

- **IAM Roles**: Assign IAM roles to EC2 instances to securely access other AWS services without embedding credentials in your application code. This enhances security by managing permissions centrally.

- **Security Groups:** Security groups Configured to control inbound and outbound traffic. Only allow necessary ports for web traffic (HTTPS) and database access (3306), minimizing exposure to potential threats.

Conclusion

Amazon EC2 gives a vigorous stage for facilitating web applications, advertising adaptability, versatility, and integration with other AWS administrations. By leveraging these capabilities, the understudy records web application can productively handle tall activity volumes whereas keeping up security and execution benchmarks.



3. RDS

Amazon RDS is utilized to oversee the database freely from the web server, giving a overseen database benefit that streamlines setup, operation, and scaling. By decoupling the database from the application server, it upgrades the in general architecture's adaptability and unwavering quality.

Security Considerations

- **Network Isolation:** The RDS instance is placed in a private subnet within the VPC, preventing direct access from the internet. This setup enhances security by limiting exposure to potential threats.
- **Access Control:** Only the web application is allowed to access the database through specific security group rules. This minimizes the risk of unauthorized access.
- **Secrets Management:** AWS Secrets Manager is used to store and manage database credentials securely. This approach avoids hardcoding sensitive information in application code.

Conclusion

We utilized Amazon RDS to give a vigorous arrangement for overseeing databases freely from application servers. By leveraging RDS's overseen administrations, the web application can accomplish superior execution, security, and operational efficiency. This setup adjusts with AWS' best hones for building adaptable and secure applications.

4. Secrets Manager And IAM

AWS Privileged insights Chief is utilized to safely store and oversee delicate data such as database accreditations. This benefit makes a difference dispose of the require to hardcode touchy information in application code, improving security by diminishing the chance of credential introduction.

Integration

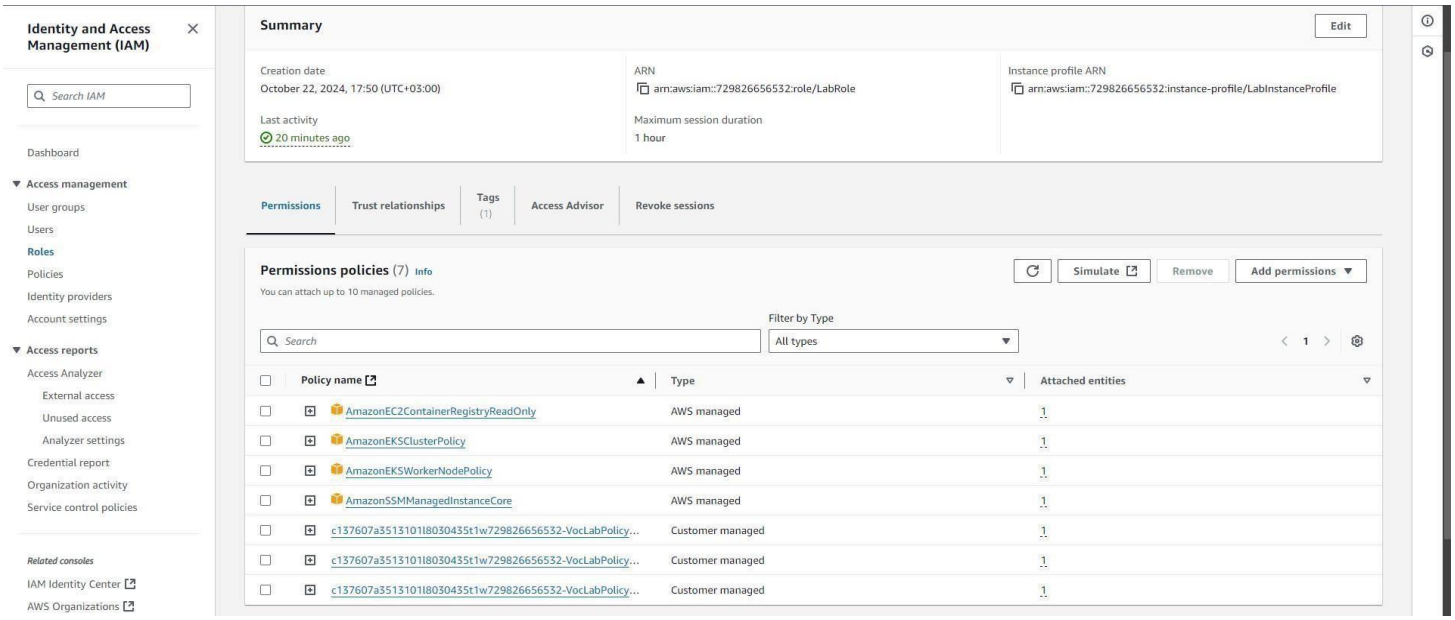
- **Access by Web Application:** The web application recovers accreditations from Privileged insights Supervisor at runtime. This approach guarantees that accreditations are not put away in the application code or setup records, minimizing security dangers.
- **IAM Roles:** The EC2 instances running the web application are assigned IAM roles with permissions to access Secrets Manager. This setup allows the application to fetch secrets securely without embedding access keys in the code.

Security Considerations

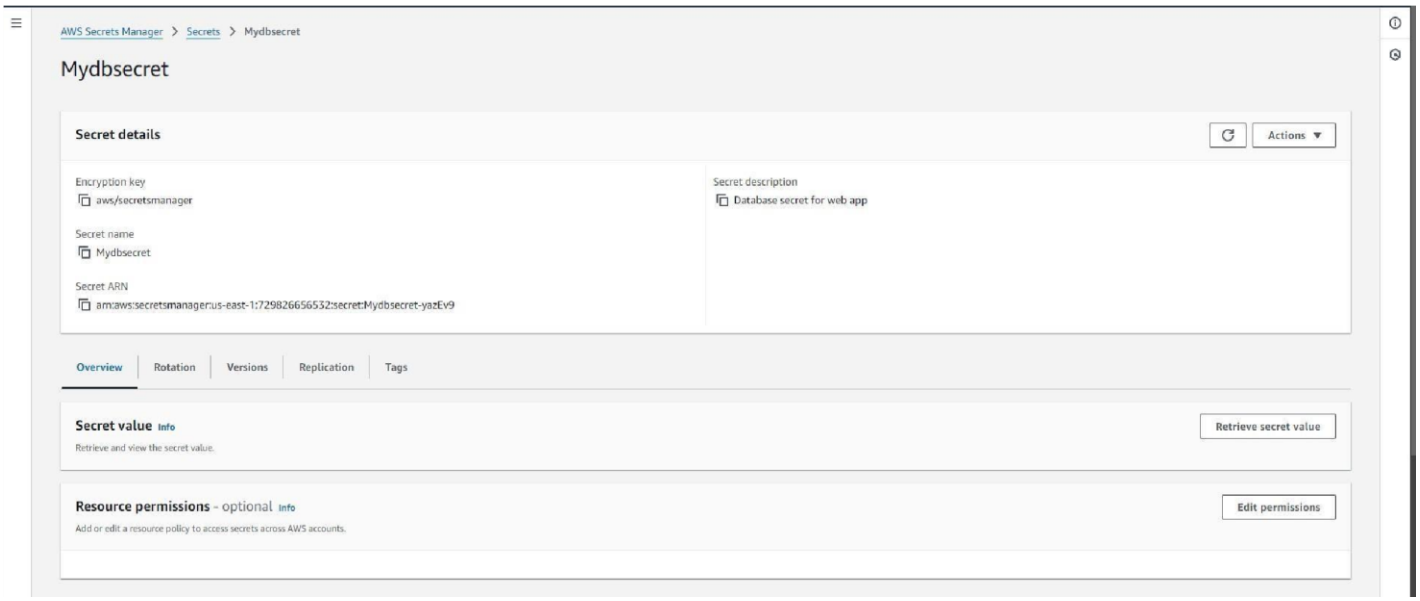
- **Least Privilege Access:** IAM policies have been configured to grant only the necessary permissions for accessing specific secrets, adhering to the principle of least privilege.
- **Network Isolation:** Ensure that access to Secrets Manager is restricted to authorized network paths within your VPC.

Conclusion

AWS Secrets Manager provides a robust solution for managing sensitive information securely. By integrating it with the web application, you ensure that database credentials are handled securely, aligning with best practices for application security and compliance.



5. Application Load Balancer



We utilized The Application Stack Balancer (ALB) to convey approaching activity over different Amazon EC2 occasions. This guarantees that no single occasion is overpowered with demands, which improves the application's accessibility and unwavering quality. By adjusting the stack, the ALB makes a difference keep up ideal execution, indeed amid crest activity periods.

Configuration

- **Listener Configuration:** The ALB uses listeners to check for connection requests from clients. We configured listeners to use HTTPS protocol, allowing for secure communication between clients and the application.
- **Target Groups**: EC2 instances are registered in target groups, which the ALB uses to route requests. This setup allows for flexible scaling and management of instances without affecting the load balancer's configuration.

Key Features

- **Health Checks:** The ALB performs health checks on registered instances to ensure that traffic is only routed to healthy instances. This feature helps maintain application uptime by automatically rerouting traffic away from unhealthy instances.
- **Security Features:** We integrated with the AWS Certificate Manager (ACM) allows for easy management of SSL/TLS certificates, enabling secure communication over HTTPS.

Security Considerations

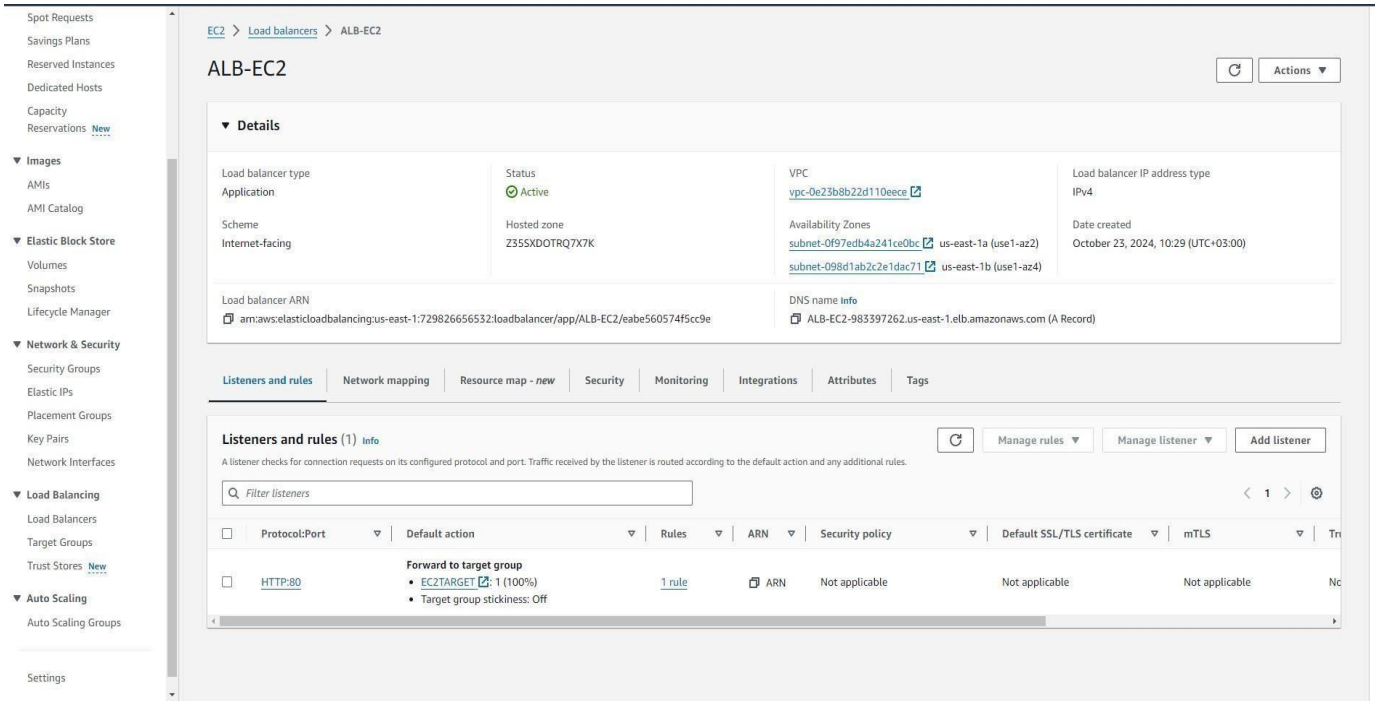
- **Security Groups:** Configure security groups for the ALB to allow only necessary inbound traffic from clients and restrict outbound traffic to specific ports used by EC2 instances.

- **IAM Policies:** Use IAM policies to control access to the ALB configuration and ensure that only authorized users can make changes.

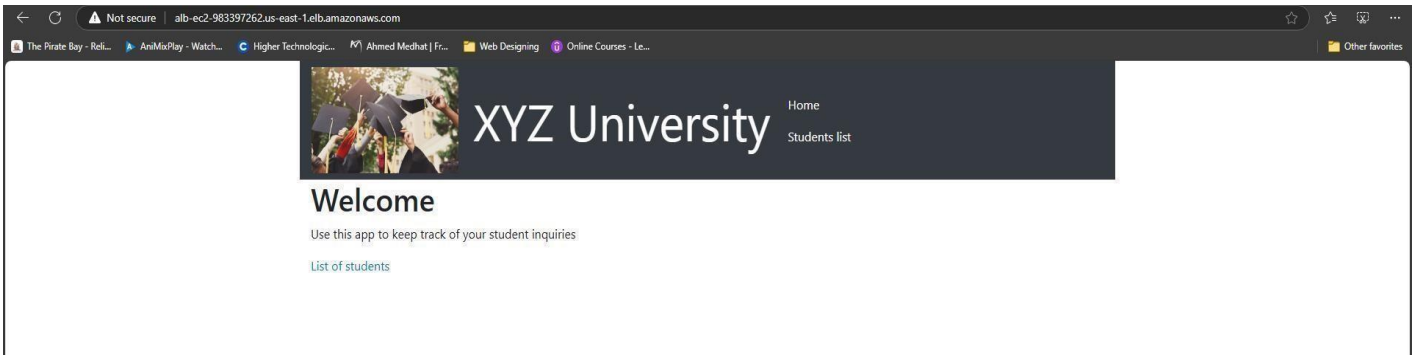
Conclusion

The Application Load Balancer plays a critical role in ensuring that the web application remains highly available

and responsive under varying loads. By distributing traffic efficiently and providing advanced routing capabilities, the ALB enhances both performance and security in accordance with AWS best practices.



6. Auto



Scaling Group

The Auto Scaling Group (ASG) automatically adjusts the number of Amazon EC2 instances based on demand. This ensures that the application maintains optimal performance by scaling out during high traffic periods and scaling in during low traffic periods, thereby optimizing resource usage and costs.

Policy

- **Target Tracking Policy:** This policy automatically adjusts the number of instances to maintain a specified metric, such as average CPU utilization. By setting a target value, the ASG ensures that the application performs efficiently under varying loads.

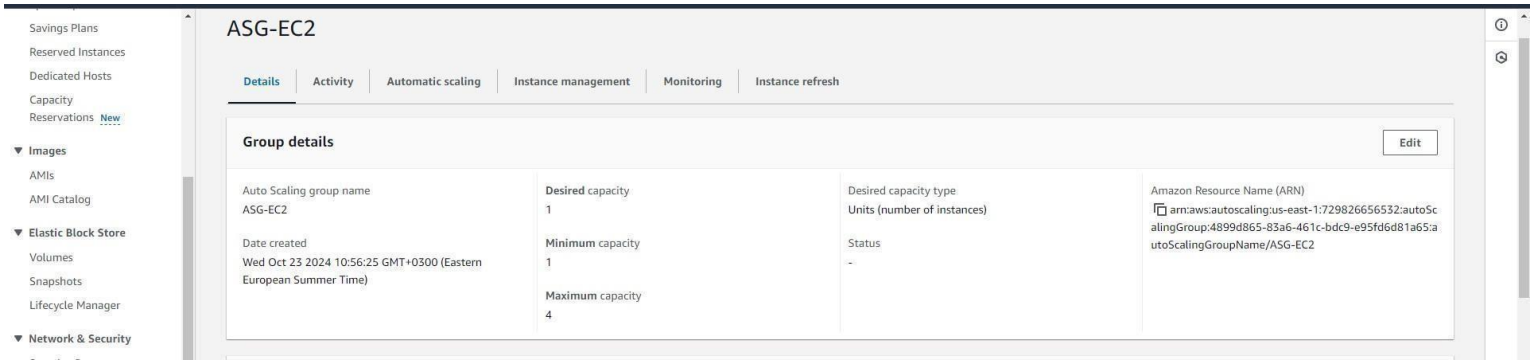
Configuration

- **Launch Template:** Defines the configuration for EC2 instances, including AMI, instance type, and security groups. This template is used by the ASG to launch and terminate instances as needed.

- **Availability Zones:** The ASG is configured to deploy instances across multiple Availability Zones to enhance fault tolerance and availability.

Conclusion

The Auto Scaling Group ensures that the web application can handle fluctuating traffic efficiently while maintaining high availability and performance. By leveraging AWS's automatic scaling capabilities, you can optimize resource usage and costs effectively.



7. Internet Gateway

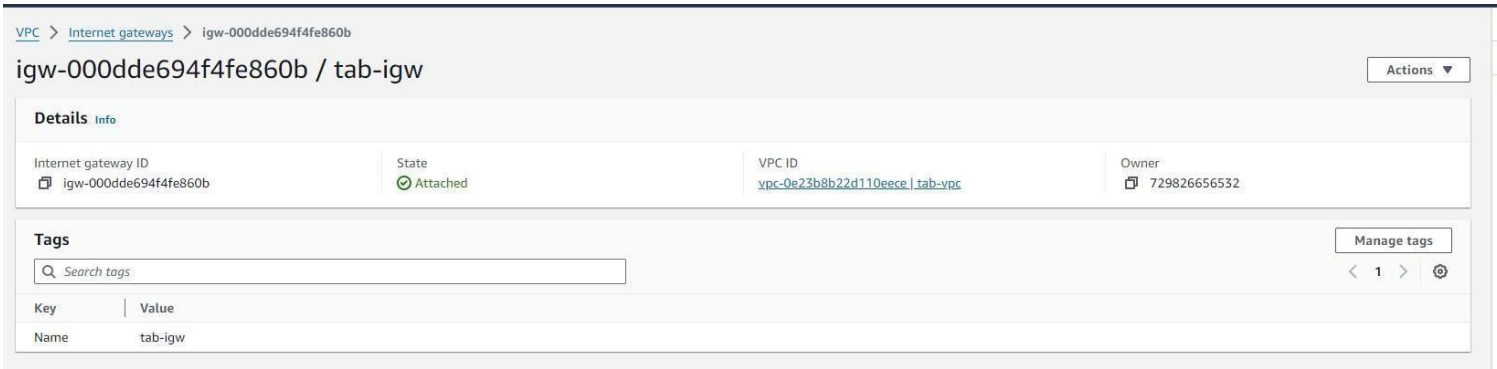
An Internet Gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It serves as a bridge to connect the VPC to the internet, enabling resources in public subnets to receive inbound traffic from the internet and send outbound traffic to the internet.

Integration with Project Components

- **Public Subnets:** The Internet Gateway is attached to the VPC and used to route traffic from public subnets. This setup allows resources such as the Application Load Balancer (ALB) and bastion hosts within these subnets to be accessible from the internet.
- **Application Load Balancer:** The ALB, which is deployed in public subnets, uses the Internet Gateway to handle incoming requests from users accessing the web application over the internet.

Conclusion

Integrating an Internet Gateway is essential for enabling internet connectivity for public-facing components of your architecture. It plays a critical role in ensuring that your web application can be accessed by users over the internet while maintaining security through proper configuration of network components.



8. Cloud9

We used AWS Cloud9 to act as a cloud-based integrated development environment (IDE) that provides a convenient platform for running AWS Command Line Interface (CLI) commands and scripts. This environment is

particularly useful for developing, testing, and deploying applications in the AWS ecosystem, as it comes preconfigured with essential tools and SDKs.

Instance Type

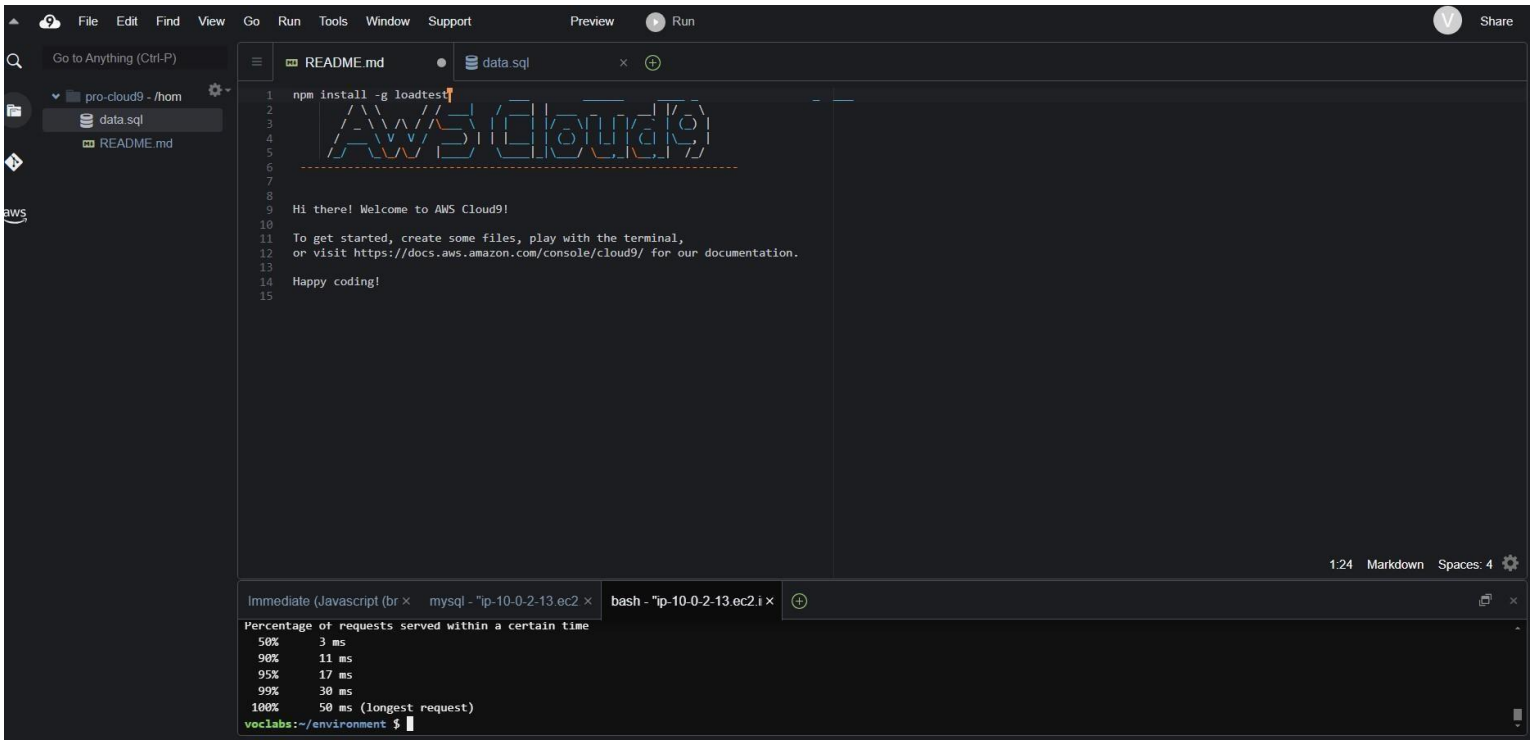
- **t3.micro:** This instance type is chosen for cost efficiency. It offers a balance of compute power and memory suitable for development tasks without incurring high costs. The t3.micro instance is part of the AWS Free Tier, making it an economical choice for development purposes.

Best Practices

- Cloud9 environment is regularly updated to ensure it has the latest security patches and software updates.
- We used the version control systems ‘Git’ within Cloud9 to manage code changes effectively.
- Monitor usage and performance metrics to optimize resource allocation and cost management.

Conclusion

AWS Cloud9 provides a robust and flexible development environment that integrates seamlessly with AWS services. By using a t3.micro instance, we developed the application efficiently while keeping costs low. The platform's features support collaborative development and streamline workflows for deploying applications in the cloud.


File Edit Find View Go Run Tools Window Support
Preview
Run
Share
Go to Anything (Ctrl-P)
pro-cloud9 - /hom
data.sql
README.md
README.md
data.sql
npm install -g loadtest
Hi there! Welcome to AWS Cloud9!
To get started, create some files, play with the terminal,
or visit https://docs.aws.amazon.com/console/cloud9/ for our documentation.
Happy coding!
1:24 Markdown Spaces: 4
Immediate (Javascript (br
mysql - "ip-10-0-2-13.ec2
bash - "ip-10-0-2-13.ec2.i
Percentage of requests served within a certain time
50% 3 ms
90% 11 ms
95% 17 ms
99% 30 ms
100% 50 ms (longest request)
voclabs:~/environment $

9. CloudWatch

We implemented Amazon CloudWatch as a monitoring and management service built for developers, system operators, site reliability engineers (SRE), and IT managers. It provides data and actionable insights to monitor applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health.

Integration with Project Components

- **EC2 Instances:** CloudWatch monitor EC2 instances for metrics such as CPU utilization, disk reads/writes, and network traffic. This helps in ensuring that the instances are performing optimally and can trigger scaling actions if needed.
- **RDS:** It monitors database performance metrics such as CPU load, memory usage, and I/O operations. This is crucial for maintaining database health and performance.
- **Application Load Balancer:** CloudWatch provides metrics on request counts, latency, and error rates for the load balancer, helping to ensure that traffic is distributed efficiently.
- **Auto Scaling:** It has been used to monitor scaling activities and ensure that the Auto Scaling policies are working as intended.
- **AWS WAF:** Monitor WAF logs and set up alerts for suspicious activities or rule violations.

Key Features

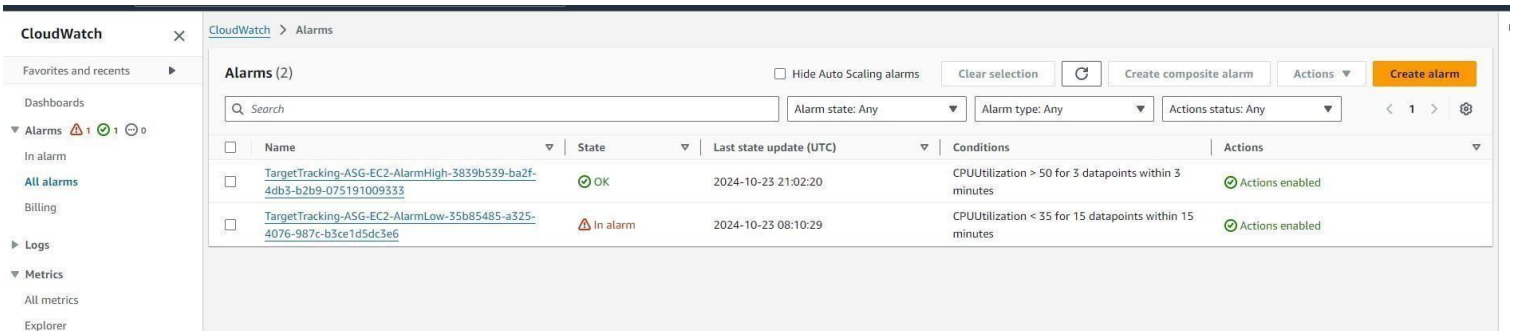
- **Alarms:** Alarms have been set up to automatically notify us of performance issues or threshold breaches, enabling proactive management.
- **Logs:** CloudWatch collect and monitor log files from EC2 instances and other AWS services. This helps in troubleshooting application issues.
- **Dashboards:** Custom dashboards have been created to visualize key metrics from different AWS services in one place for easy monitoring.

Benefits

- **Real-Time Monitoring:** Provides real-time insights into application performance and resource utilization.
- **Automated Responses:** Integrates with AWS Lambda to automate responses to specific conditions.
- **Cost Management:** Helps in optimizing resource usage by providing insights into underutilized resources.

Conclusion

Amazon CloudWatch is a comprehensive monitoring tool that integrates seamlessly with the AWS services used in this project. It provides the necessary insights and tools to ensure that the web application remains highly available, scalable, and high performing.



1. Developing a Cost Estimate

- Utilize the AWS Estimating Calculator to gauge the taken a toll of running the arrangement in the us-east1 Locale for 12 months.

Estimate summary

Upfront cost	Monthly cost	Total 12 months cost
318.64 USD	152.53 USD	2,149.00 USD Includes upfront cost

Detailed Estimate

Name	Group	Region	Upfront cost	Monthly cost
Amazon EC2	No group applied	US East (N. Virginia)	122.64 USD	0.00 USD
Status: - **Description:** **Config summary:** Tenancy (Shared Instances), Operating system (Linux), Workload (Consistent, Number of instances: 2), Advance EC2 instance (t3.micro), Pricing strategy (Compute Savings Plans 1yr All Upfront), Enable monitoring (disabled), DT Inbound: Not selected (0 TB per month), DT Outbound: Not selected (0 TB per month), DT Intra-Region: (0 TB per month)				
Amazon RDS for MySQL	No group applied	US East (N. Virginia)	196.00 USD	40.30 USD
Status: - **Description:** **Config summary:** Storage amount (80 GB), Storage for each RDS instance (General Purpose SSD (gp2)), Nodes (1), Instance type (db.t3.micro), Utilization (On-Demand only) (50 %Utilized/Month), Deployment option (Multi-AZ), Pricing strategy (Reserved 1yr All Upfront)				
Elastic Load Balancing	No group applied	US East (N. Virginia)	0.00 USD	45.63 USD
Status: - **Description:** **Config summary:** Number of Application Load Balancers (1)				

Conclusion

This execution arrange gives a organized approach to sending a strong web application on AWS, guaranteeing it meets all useful, security, versatility, and giving an moved forward client encounter amid top periods.

Amazon Virtual Private Cloud (VPC)	No group applied	US East (N. Virginia)	0.00 USD	66.60 USD

Status: -
Description:
Config summary: Number of NAT Gateways (2)