AWS Multi-Region Resilience Monitor: Technical Documentation

By: Ritvik Indupuri

Date: 10/25/2025

Table of Contents

Introduction

This document provides a comprehensive technical overview of the AWS Multi-Region Resilience and Outage Prevention System. The system is designed to provide proactive monitoring, predictive outage detection, and automated failover for multi-region AWS environments. This document is intended for engineers, architects, and developers who are responsible for maintaining and extending the system.

- **Purpose**

This document provides a technical overview of the AWS Multi-Region Resilience Monitor. The system is designed for proactive monitoring of critical DNS endpoints in multi-region AWS environments, incorporating anomaly detection and automated failover capabilities to mitigate potential outages.

- **Target Audience**

This documentation is intended for engineers, architects, and developers responsible for deploying, maintaining, or extending this monitoring and resilience system.

System Architecture

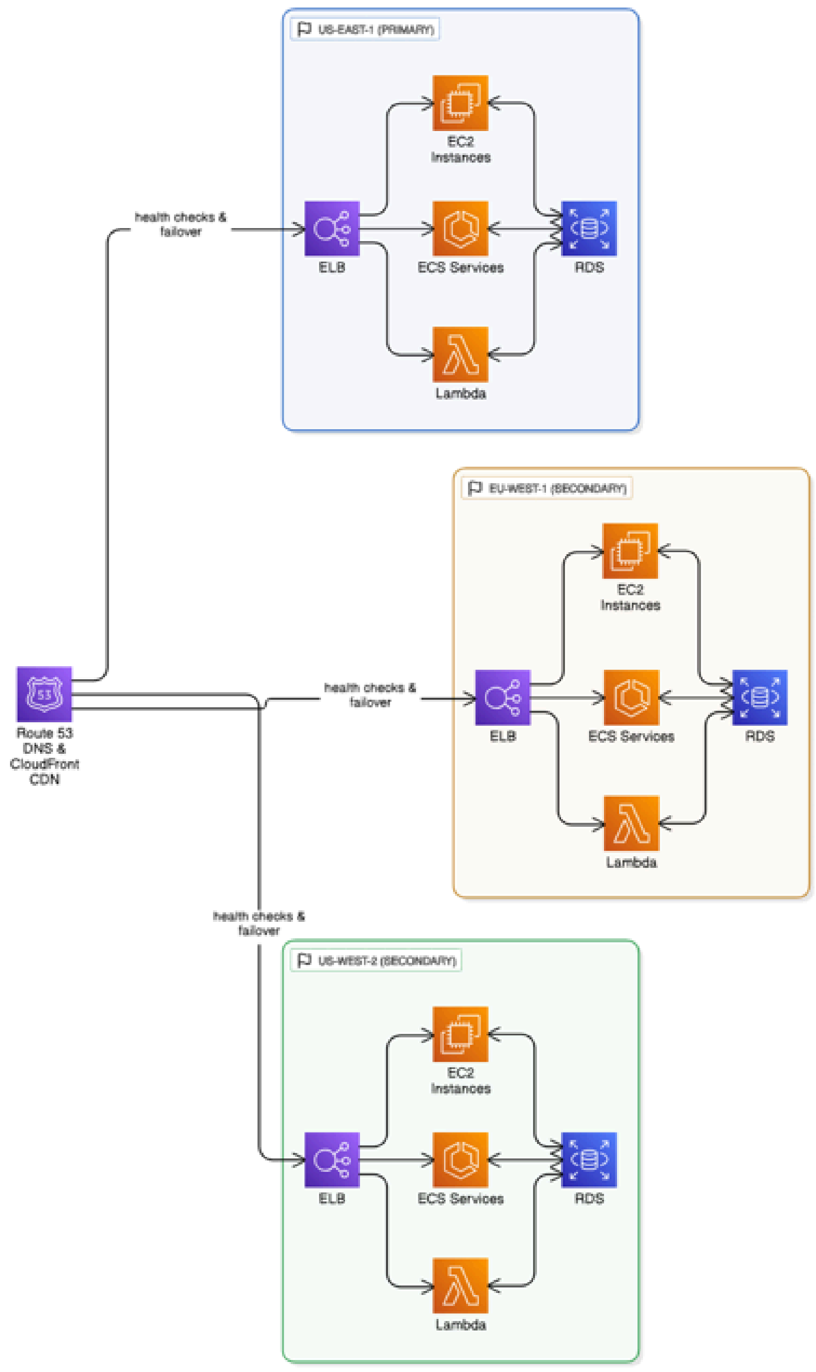


Figure 1: High-level architecture showing multi-region deployment and monitoring components.

The system is composed of several key components that work together to provide a comprehensive solution for multi-region resilience and outage prevention. The architecture is designed to be modular and scalable, allowing for easy extension and customization.

The core of the system is the Outage Prevention System, which is responsible for monitoring the health of the AWS environment, predicting potential outages, and triggering automated failover procedures. This system is composed of the following sub-components:

- Enhanced DNS Monitor: This component is responsible for monitoring the health of the DNS records for the multi-region environment. It periodically performs DNS lookups and measures the response times for each of the configured endpoints.
- Multi-Region Failover: This component is responsible for executing the automated failover procedures. It is triggered by the Outage Prevention System when a potential outage is detected. The failover process involves updating the DNS records to redirect traffic to the healthy region.
- Web Dashboard: This component provides a real-time view of the health of the multi-region environment. It displays the status of the DNS records, the response times for each of the configured endpoints, and the overall health of the system.
- Machine Learning Model: This component is responsible for predicting potential outages based on the historical data collected by the Enhanced DNS Monitor. It uses a machine learning model to identify patterns in the data that are indicative of an impending outage.

Web Dashboard

Dashboard Overview

The web dashboard provides a real-time, at-a-glance view of the health of the multi-region AWS environment. It is designed to be intuitive and easy to understand, allowing for quick identification of potential issues.

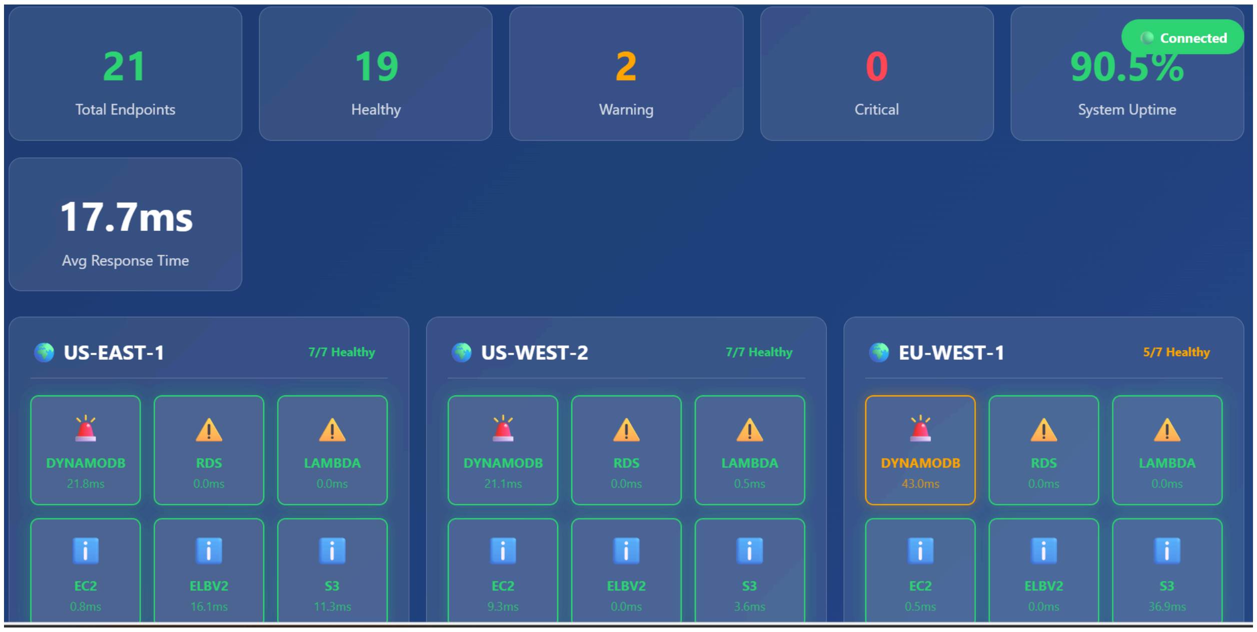


Figure 2: Real-time web dashboard displaying multi-region AWS service health.*

The dashboard is divided into two main sections:

1. Summary Metrics: A top-level overview of the entire system's health.
2. Regional Metrics: A detailed breakdown of the health of each AWS region being monitored.

Data Sources and Metrics

All data displayed on the dashboard is generated by the `web_dashboard.py` script. This script performs DNS lookups for a predefined set of AWS service endpoints across multiple regions, measures the response times, and aggregates the data for display. The dashboard updates every 10 seconds with fresh data pushed via WebSockets.

Summary Metrics

The summary metrics at the top of the dashboard provide a high-level overview of the system's health.

- Total Endpoints: The total number of AWS service endpoints being monitored across all regions.
- Healthy: The number of endpoints that are currently in a 'healthy' state.
- Warning: The number of endpoints that are currently in a 'warning' state.
- Critical: The number of endpoints that are currently in a 'critical' state.
- System Uptime: This is a calculated metric representing the percentage of healthy endpoints, calculated as `(Healthy Endpoints / Total Endpoints) * 100`. The "Connected" indicator signifies an active WebSocket connection to the backend.
- Avg Response Time: The average response time in milliseconds across all monitored endpoints.

Regional Metrics

The dashboard provides a card for each monitored AWS region (e.g., `us-east-1`, `us-west-2`, `eu-west-1`), showing a detailed health breakdown.

- Region Health: The header of each card shows the number of healthy endpoints out of the total for that region (e.g., "7/7 Healthy").
- Service Status: Each card contains a set of smaller cards, one for each monitored AWS service (e.g., DynamoDB, RDS, EC2).
  - Response Time: The time displayed for each service is the average response time for all of that service's endpoints within that region.

    - Status Icon and Color: The icon and border color of the service card indicate its overall status. The status is determined by the "worst" status of any of its endpoints. For example, if even one of a service's endpoints is in a 'critical' state, the entire service card will be marked as 'critical'.

Status Determination (Warning and Critical)

The 'healthy', 'warning', and 'critical' statuses for each individual endpoint are determined by its DNS lookup response time.

- Default Thresholds:
    - Healthy: Response time <= 100ms
    - Warning: Response time > 100ms and <= 200ms
    - Critical: Response time > 200ms or a failed DNS lookup.
- Stricter Thresholds for Critical Services:
    - Services marked as 'critical' priority, such as DynamoDB, have stricter thresholds to ensure early detection of potential issues.
    - DynamoDB Thresholds:
        - Healthy: Response time <= 50ms
        - Warning: Response time > 50ms and <= 100ms
        - Critical: Response time > 100ms or a failed DNS lookup.

This tiered approach to status determination allows the system to prioritize alerts and draw attention to issues with the most critical components of the infrastructure.

System Health Trends

The dashboard also features a "System Health Trends" graph, which provides a historical view of the system's health over time.

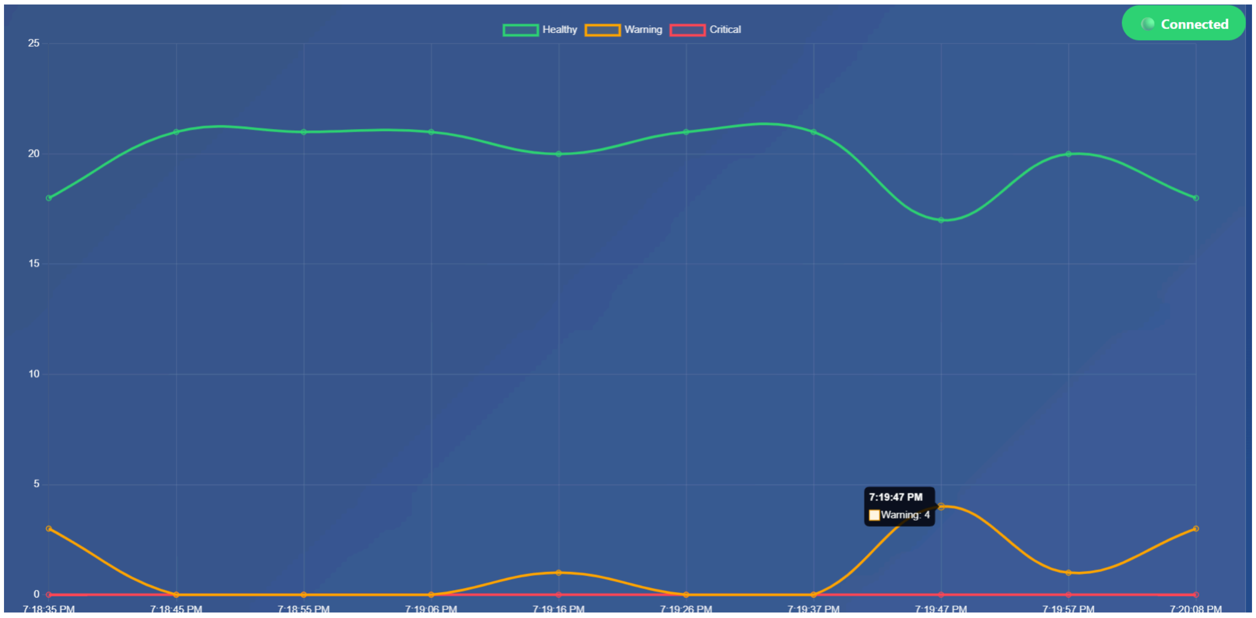


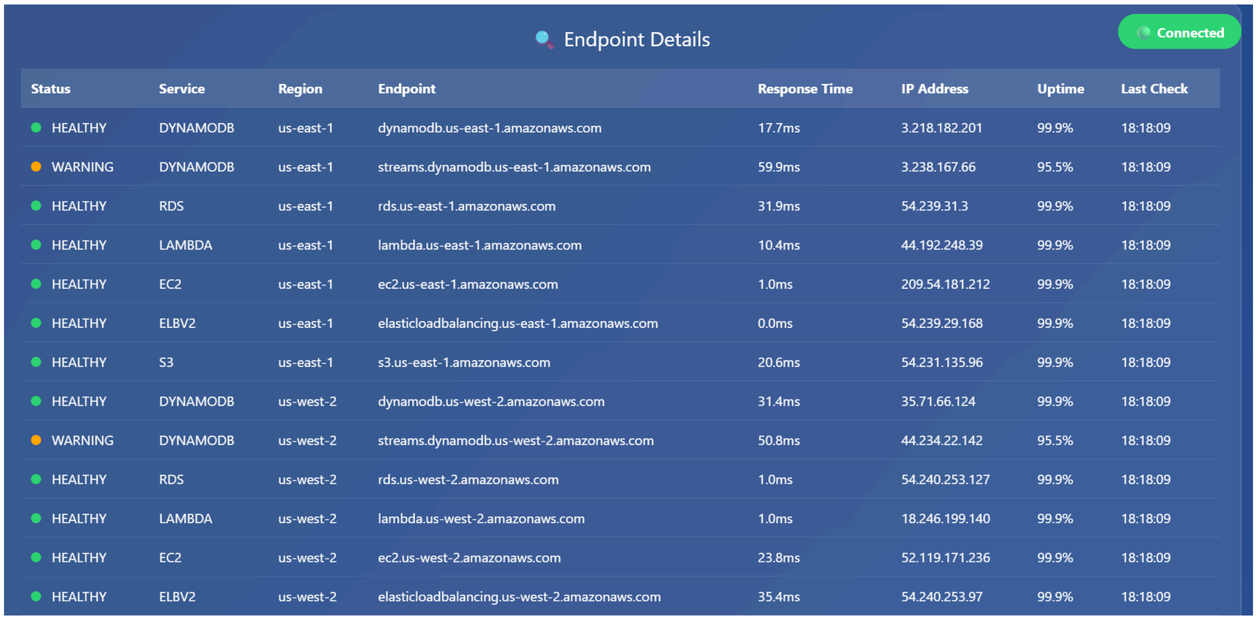
Figure 3: Historical graph showing endpoint status (Healthy, Warning, Critical) over time.

This graph displays the number of endpoints in each of the 'healthy', 'warning', and 'critical' states over the last several monitoring intervals. This allows for easy identification of trends in the system's health, such as a gradual increase in the number of 'warning' or 'critical' endpoints, which could be an early indicator of a developing issue.

- Data Source: The data for this graph is collected by the perform_health_checks function in the web_dashboard.py script. With each 10-second monitoring interval, the script records the number of endpoints in each state ('healthy', 'warning', 'critical') and stores this data in a historical log. The dashboard then displays the last 20 data points from this log, providing a rolling window of the system's recent health.
- X-Axis: The x-axis of the graph represents time, with each data point corresponding to a monitoring interval.
- Y-Axis: The y-axis represents the number of endpoints in each state.
- Tooltip: The tooltip that appears when hovering over a data point provides a snapshot of the system's health at that specific time. For example, the tooltip in the screenshot, "Warning: 4", indicates that at 7:19:47 PM, there were 4 endpoints in the 'warning' state.

Endpoint Details Table

The main table on the dashboard provides a detailed, real-time status for every individual AWS service endpoint being monitored.



Status	Service	Region	Endpoint	Response Time	IP Address	Uptime	Last Check
● HEALTHY	DYNAMODB	us-east-1	dynamodb.us-east-1.amazonaws.com	17.7ms	3.218.182.201	99.9%	18:18:09
● WARNING	DYNAMODB	us-east-1	streams.dynamodb.us-east-1.amazonaws.com	59.9ms	3.238.167.66	95.5%	18:18:09
● HEALTHY	RDS	us-east-1	rds.us-east-1.amazonaws.com	31.9ms	54.239.31.3	99.9%	18:18:09
● HEALTHY	LAMBDA	us-east-1	lambda.us-east-1.amazonaws.com	10.4ms	44.192.248.39	99.9%	18:18:09
● HEALTHY	EC2	us-east-1	ec2.us-east-1.amazonaws.com	1.0ms	209.54.181.212	99.9%	18:18:09
● HEALTHY	ELBV2	us-east-1	elasticloadbalancing.us-east-1.amazonaws.com	0.0ms	54.239.29.168	99.9%	18:18:09
● HEALTHY	S3	us-east-1	s3.us-east-1.amazonaws.com	20.6ms	54.231.135.96	99.9%	18:18:09
● HEALTHY	DYNAMODB	us-west-2	dynamodb.us-west-2.amazonaws.com	31.4ms	35.71.66.124	99.9%	18:18:09
● WARNING	DYNAMODB	us-west-2	streams.dynamodb.us-west-2.amazonaws.com	50.8ms	44.234.22.142	95.5%	18:18:09
● HEALTHY	RDS	us-west-2	rds.us-west-2.amazonaws.com	1.0ms	54.240.253.127	99.9%	18:18:09
● HEALTHY	LAMBDA	us-west-2	lambda.us-west-2.amazonaws.com	1.0ms	18.246.199.140	99.9%	18:18:09
● HEALTHY	EC2	us-west-2	ec2.us-west-2.amazonaws.com	23.8ms	52.119.171.236	99.9%	18:18:09
● HEALTHY	ELBV2	us-west-2	elasticloadbalancing.us-west-2.amazonaws.com	35.4ms	54.240.253.97	99.9%	18:18:09

Figure 4: Detailed table view showing the status, response time, and IP address for each monitored endpoint.

This table is the most granular view of the system's health and is designed to allow for quick identification of specific problem areas.

- Status: A color-coded indicator (green for 'healthy', orange for 'warning', red for 'critical') that provides an immediate visual cue of the endpoint's health.
- Service: The AWS service to which the endpoint belongs (e.g., DYNAMODB, RDS, EC2).
- Region: The AWS region where the endpoint is located (e.g., us-east-1).
- Endpoint: The full DNS hostname of the service endpoint.
- Response Time: The time taken, in milliseconds, to resolve the DNS for this specific endpoint.
- IP Address: The IP address that the endpoint's DNS resolved to.
- Uptime: A simulated uptime percentage based on the endpoint's current status. For a 'healthy' status, this is 99.9%; for 'warning', it is 95.5%; and for 'critical', it is 85.2%.

- Last Check: The time at which the last health check was performed for this endpoint.

Machine Learning Model

The system utilizes a machine learning model to proactively detect anomalies in DNS health check data, which could be early indicators of systemic issues or impending outages.

- Model: The system employs the IsolationForest algorithm from the scikit-learn library. This is an unsupervised anomaly detection model, which is well-suited for this use case as it can identify unusual patterns without requiring pre-labeled training data of past outages.
- Purpose: The primary purpose of the model is to identify subtle, widespread anomalies across the monitored DNS endpoints that might not be caught by simple threshold-based alerting. For example, a slight increase in response times across multiple services and regions might not trigger individual 'warning' or 'critical' alerts, but could be indicative of a larger, underlying problem.
- Training and Prediction: The model is not pre-trained. Instead, it is trained on-the-fly with the most recent set of DNS health check data. The fit_predict method is used to both train the model on the current data and identify any anomalies within that same dataset. This allows the model to continuously adapt to the latest performance patterns of the system.
- Data Features: The model is trained on the following features from the DNS health check data:
    i. success: A binary value indicating whether the DNS resolution was successful (1) or not (0).
    ii. response_time: The response time of the DNS query in milliseconds.
    iii. hour: The hour of the day when the check was performed, which allows the model to account for time-based patterns in system performance.
- Anomaly Detection and Alerting:
    i. The system collects a batch of the latest DNS health checks.
    ii. The data is preprocessed using StandardScaler to normalize the feature data.
    iii. The IsolationForest model is then used to identify anomalies in the data.
    iv. If the number of detected anomalies exceeds a certain threshold (currently 10% of the data points in the batch), a 'high' severity alert is generated.
    v. This alert is then sent to the automated response system, which can trigger notifications and other preventative actions.

System Components

Enhanced DNS Monitor

The Enhanced DNS Monitor is the first line of defense, responsible for high-frequency health checks of critical AWS service endpoints.

- Purpose: To continuously monitor the availability and performance of AWS DNS endpoints.
- Functionality:
    - Performs DNS lookups against a predefined list of service endpoints (e.g., dynamodb.us-east-1.amazonaws.com).
    - Measures and records the response time for each lookup.
    - Identifies and logs DNS resolution errors.
    - Operates independently of AWS SDKs for its core monitoring functions, ensuring it remains operational even during AWS control plane issues.

Outage Prevention System

This component acts as the central intelligence of the system, responsible for analyzing data, predicting failures, and initiating responses.

- Purpose: To analyze monitoring data, assess the risk of cascading failures, and trigger automated responses.
- Functionality:
    - Cascade Risk Analysis: Utilizes a predefined map of service dependencies to calculate a "cascade risk score." This score quantifies the potential impact of a single service's failure on other parts of the infrastructure.
    - Alerting: Generates detailed outage alerts with severity levels, predicted impact, and recommended actions.
    - Automated Response Trigger: Initiates the Multi-Region Failover Manager when a critical, high-risk event is detected. The system has a heightened sensitivity to DynamoDB DNS failures, reflecting its core design inspiration.

Multi-Region Failover Manager

The Multi-Region Failover Manager is the automated, action-oriented component that executes the regional failover process.

- Purpose: To automatically and gracefully shift traffic from an unhealthy region to a healthy one.
- Functionality:
    - Deep Health Assessment: Performs comprehensive health checks within each region, evaluating the status of specific services like ELB, EC2, and RDS to calculate a regional "health score."
    - Automated Failover Process:
        a. Selects the best failover region based on health score and response time.
        b. Updates Route 53 to redirect DNS traffic.
        c. Updates CloudFront distributions to use the new region's origins.
        d. Scales up resources (Auto Scaling Groups, ECS services) in the new region.
        e. Verifies that the new region is healthy and ready to handle traffic before completing the failover.
    - Automatic Rollback: If the failover process fails, the system will attempt to automatically revert the changes.

Getting Started

Prerequisites

- Python 3.8+
- An AWS account with programmatic access (API key and secret)
- Your AWS credentials configured in your environment (e.g., via `~/.aws/credentials` or environment variables)

Installation

1. Clone the repository:

```
git clone <repository-url>
```

2.
```
cd <repository-directory>
```
3. Install the required Python packages:
4.
```
pip install -r requirements.txt
```

Running the System

The system's components are designed to be run as standalone scripts. It is recommended to run them in separate terminal sessions to observe their individual outputs.

1. Enhanced DNS Monitor:
2.
```
python enhanced_dns_monitor.py
```
3. Outage Prevention System:
4.
```
python outage_prevention_system.py
```
5. Multi-Region Failover Manager:
6.
```
python multi_region_failover.py
```
7. Web Dashboard:
8.
```
python web_dashboard.py
```
9. The dashboard will be available at `http://localhost:5000`.

Testing

The project does not currently include a dedicated test suite. However, each of the main application scripts (enhanced_dns_monitor.py, outage_prevention_system.py, multi_region_failover.py) includes a demonstration of its core functionality within an if __name__ == "__main__" block.

To test the system, you can run these scripts individually as described in the "Getting Started" section. This will allow you to observe the monitoring and alerting features in a controlled environment.

- enhanced_dns_monitor.py: Running this script will start the DNS monitoring process and print the results of the health checks to the console.
- outage_prevention_system.py: This script will start the outage prevention system, which will perform DNS health checks, analyze the results for potential failures, and print any generated alerts to the console.
- multi_region_failover.py: This will start the failover manager, which will continuously monitor the health of the configured AWS regions and print the status to the console.
- web_dashboard.py: Running this script will launch the web dashboard, allowing you to visually inspect the health of the monitored endpoints.

Conclusion

The AWS Multi-Region Resilience Monitor offers a practical framework for enhancing the availability of multi-region applications on AWS. By integrating continuous DNS health checks, machine learning-based anomaly detection, and automated failover logic, the system provides early detection of potential DNS-related issues that could lead to broader service disruptions. Its modular design facilitates customization and integration, making it a valuable component for improving the overall resilience and reliability of critical AWS infrastructure.