**Dynamic WordPress Instance with Autoscaling and Route 53 Monitoring**

**Project Abstract:**

This project entails implementing a dynamic WordPress instance with an autoscaling solution, bolstered by Route 53 monitoring. This solution ensures optimal performance and availability by automatically adjusting server capacity based on traffic demands. Leveraging Route 53's monitoring capabilities, we'll maintain constant vigilance over instance health, swiftly identifying and resolving any potential issues to uphold seamless user experiences.

**Services Used:**

- AWS CloudFormation automates the deployment and management of resources in an AWS environment using declarative templates.

- Amazon EC2 (Elastic Compute Cloud) for virtual machines.

- Amazon Auto Scaling for automatically adjusting the number of instances based on traffic.

- Route 53 Health Monitoring enables continuous monitoring and management of the health and performance of resources and applications on AWS.

**Project Details Steps (as Completed by Me):**

1. Infrastructure Design:
   * Use AWS CloudFormation to create a template for the WordPress instance, defining resources like EC2 instances, load balancers, and auto scaling groups.
   * Design the architecture to include auto scaling groups across multiple availability zones for high availability.
   * Specify the instance type, security groups, key pairs, and other configurations for the EC2 instances.
   * Configure health checks and monitoring using Route 53 Health Monitoring to ensure continuous monitoring of instance health.
2. AWS CloudFormation Deployment:
   * Log in to the AWS Management Console and navigate to the CloudFormation service.
   * Create a new stack and upload the CloudFormation template created in the previous step.
   * Provide necessary parameters such as instance type, key pair, and other configuration settings.
   * Review the stack details and deploy the CloudFormation stack.

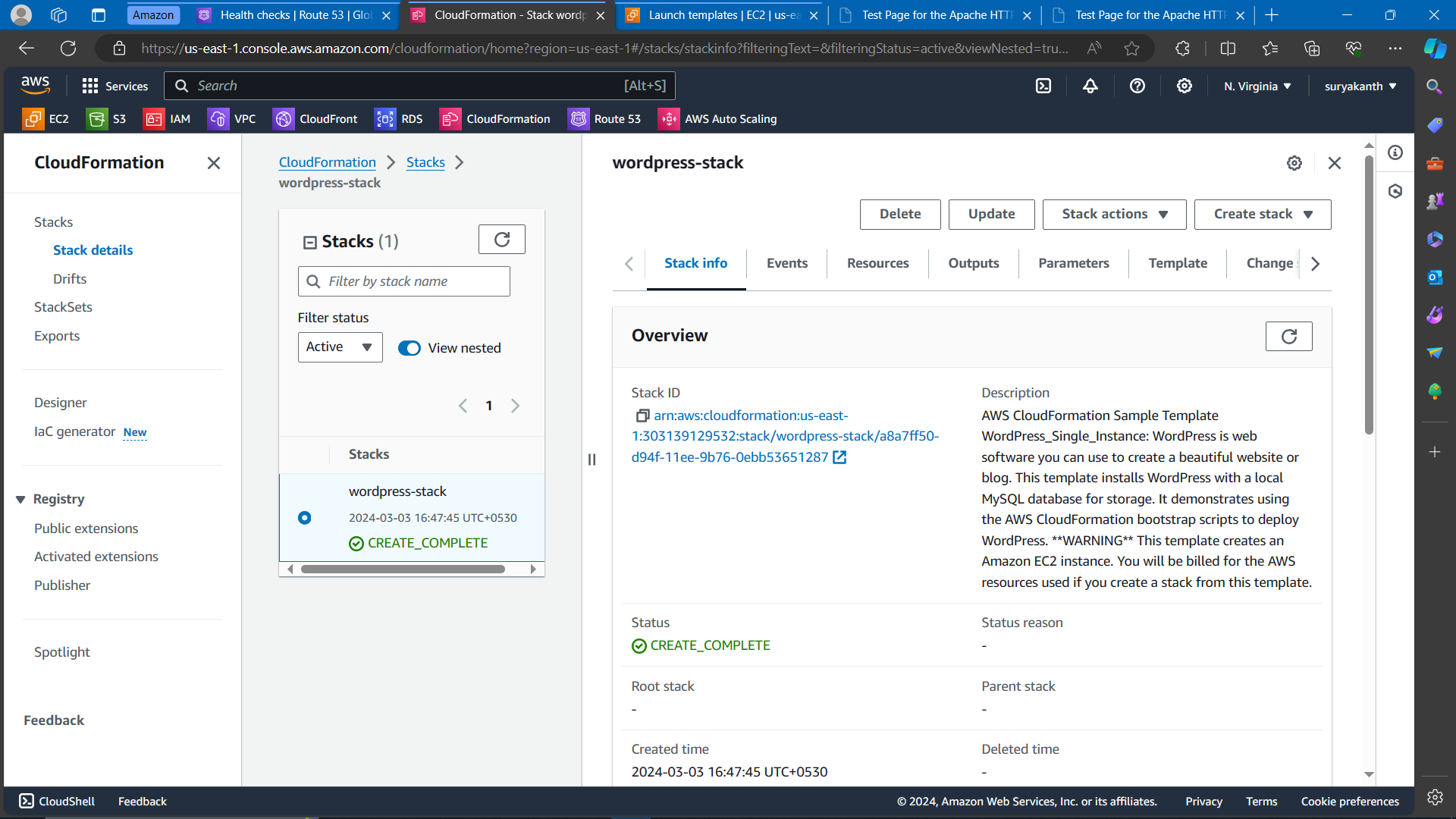


Fig 1 – Wordpress configuration using CloudFormation using sample template

1. Amazon EC2 Configuration:
   * Access the EC2 dashboard and ensure that the instances launched by the CloudFormation stack are running and accessible.
   * Configure the WordPress instance by installing the necessary software and dependencies.
   * Customize the WordPress installation according to project requirements, including themes, plugins, and content.

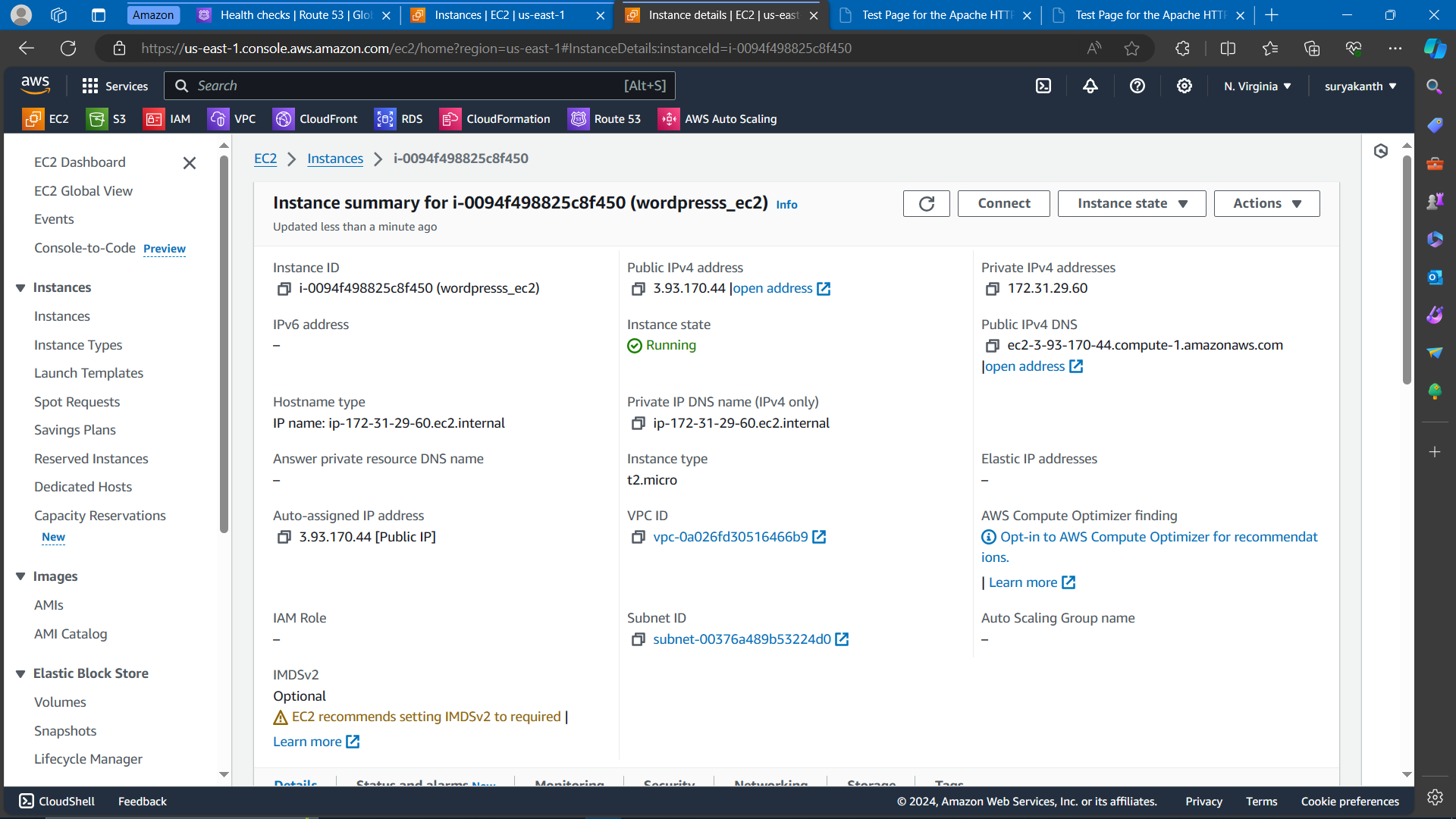
****

Fig 2 – EC2 configuration of wordpress instance

1. Amazon Auto Scaling Configuration:
   * Access the Auto Scaling dashboard and create an auto scaling group for the WordPress instances.
   * Define scaling policies based on metrics like CPU utilization or request count to automatically adjust the number of instances.
   * Set up scaling triggers and cooldown periods to ensure smooth scaling operations without causing instability.

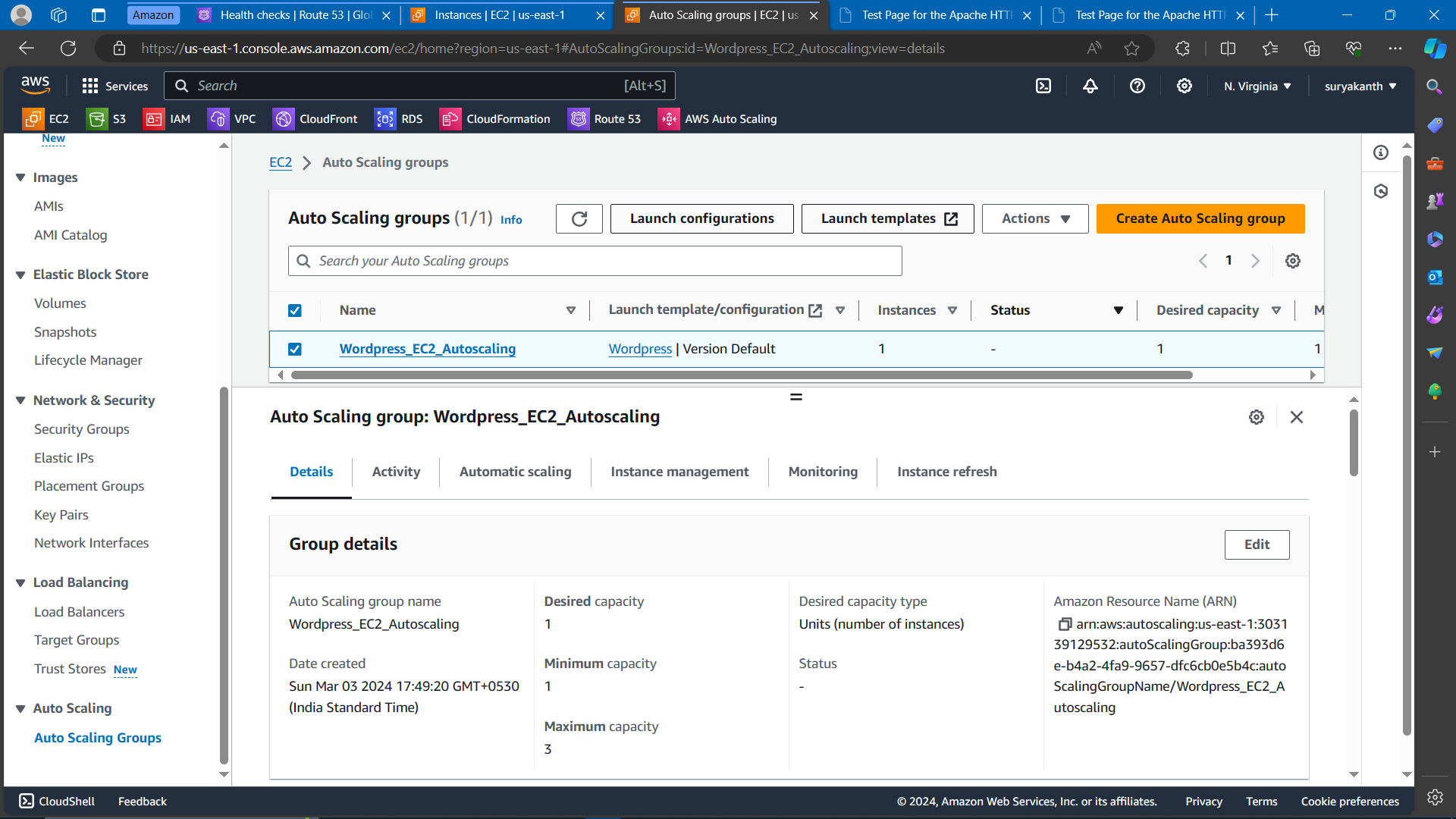
****

Fig 3 – Autoscaling configuration to balance traffic

1. Route 53 Health Monitoring Setup:
   * Navigate to the Route 53 dashboard and create health checks for the WordPress instances.
   * Define endpoints to monitor, such as the WordPress website URL, and set the monitoring interval.
   * Configure Route 53 to automatically route traffic away from unhealthy instances based on the health check results.

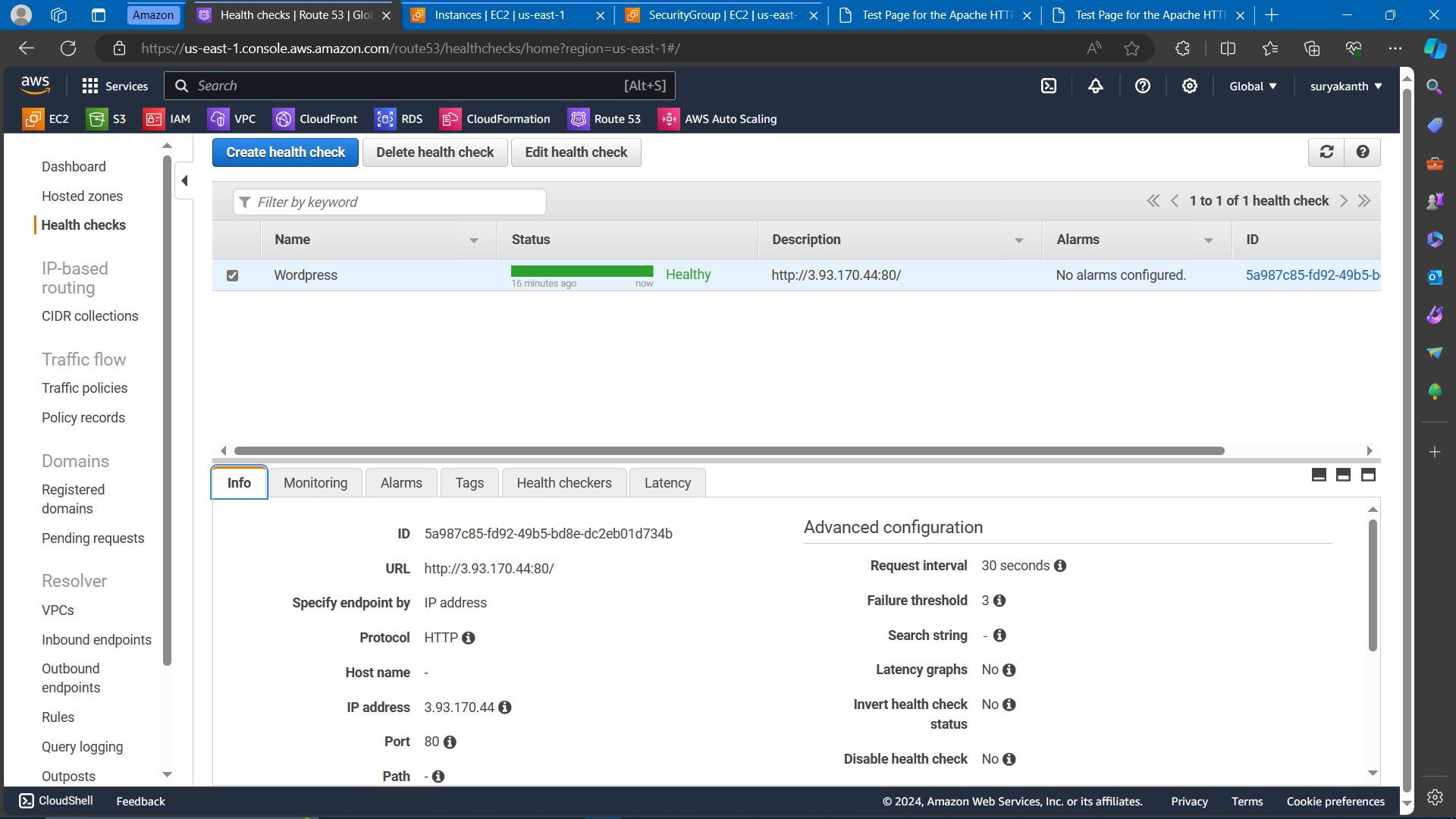
****

Fig 4 – Route 53 health check monitoring

1. Testing and Optimization:
   * Perform thorough testing to ensure that auto scaling works as expected under various traffic conditions.
   * Monitor the performance of the WordPress instances and auto scaling activities using CloudWatch metrics.
   * Optimize auto scaling configurations and thresholds based on testing results to achieve optimal performance and cost efficiency.
2. Monitoring and Maintenance:
   * Set up CloudWatch alarms to receive notifications for any performance issues or scaling events.
   * Regularly monitor instance health and performance using CloudWatch dashboards and logs.
   * Perform routine maintenance tasks such as software updates, security patches, and instance resizing as needed to maintain optimal performance and availability**.**

**Project Delivery Note:**

The project was successfully completed. The implemented solution demonstrated robustness and high availability, ensuring seamless user experience even during high traffic scenarios. The design allowed for automatic scaling of resources based on demand, enhancing cost-efficiency.