SETTING UP A WEBSITE ON AWS CLOUD

Actions Taken :

1. EC2 Instance Setup with IIS and Static Website :-

I initiated the deployment by launching an EC2 instance using the "Microsoft Windows Server 2012 R2 Base" AMI on AWS. Leveraging user data scripts, I automated the installation of IIS (Internet Information Services) during instance initialization., I uploaded a static website to the instance's directory (`C:\inetpub\wwwroot\`) and verified its functionality using the instance's public IP address.

2. Creation of Amazon Machine Image (AMI) :-

Once the static website deployment and IIS configuration were confirmed, I created an Amazon Machine Image (AMI). This AMI captured the configured state of the EC2 instance, including installed software and settings. By saving this snapshot, I established a blueprint for future deployments, facilitating consistent and efficient replication of the web server environment.

3. Launching a New EC2 Instance Using the AMI :-

Utilizing the created AMI, I launched a new EC2 instance on AWS. This step ensured that the new instance inherited the exact configuration of IIS and the deployed static website as encapsulated in the AMI. By leveraging AMIs, I maintained consistency across instances and simplified the process of scaling up or replacing instances as needed.

4. Configuration of Target Group for Load Balancing :-

To optimize scalability and ensure fault tolerance, I configured a target group within AWS. Both the original EC2 instance and the newly launched instance were added to this target group. This setup facilitated load balancing, enabling AWS to distribute incoming traffic evenly across instances based on defined criteria such as round-robin or least connections.

5. Setup of Application Load Balancer (ALB) :-

For effective traffic distribution and improved availability, I implemented an Application Load Balancer (ALB) on AWS. The ALB served as a centralized hub for managing incoming HTTP traffic to the instances within the target group. By configuring listeners and routing rules, the ALB efficiently directed traffic, enhancing overall website performance and responsiveness . I verified the operational status and accessibility of the static website by accessing it through the DNS name associated with the ALB, confirming that the load balancer correctly routed traffic to the deployed instances..

Conclusion:-

By following this structured approach, I successfully deployed and configured a scalable web server infrastructure on AWS. This methodology leverages automation for rapid deployment, AMIs for consistency in configuration, and load balancing for enhanced performance and fault tolerance. The setup ensures the client's website operates seamlessly in the cloud environment, meeting operational demands and accommodating fluctuating levels of web traffic effectively.