En primera instancia , todas las request de DNS van a pasar por Route 53, donde están configurados las hosted zone, el dominio de la web y los records. Tiene un AWS Shield para mitigar ataques DdoS para proteger la app que està corriendo.
Utilicé un AWS WAF para habilitar o bloquear request , para brindarle seguridad al cloudfront y agregar una protección extra de ataques de ataques SQL injection o XSS.

Uso un cloudfront para usar el CDN de AWS , para replicar en cache en todos los edges alrededor del mundo, brindando una excelente respuesta, no importa en que zona estè el usuario final y está conectado a un S3 Bucket, que alacena todo el contenido estático de mi web. Esta metodología ayuda a disminuir las operaciones que llegan nuestra app, mejorando nuestra performance y que tenga menos carga.

Para tolerancia a fallos, la app està replicada en dos A.Z. para obtener alta redundancia. Las operaciones que llegan desde el Cloudfront, son recibida primero por el external ELB que distribuye la carga entre las AZ, pero primero pasan por el Network ACLS , otro firewal a nivel subnets con sus propias reglas inbound y outbond. Si está habilitado, pasa los 2 grupos de EC2 instances que corren mi web server , que tienen configurado Autoescaling, dependiendo de la demanda de trafico recibido, crecen o decrecen horizontalmente. Estàn dentro de un security group para brindar control y seguridad, es un firewall interno de la EC2.

Mis request pasan a un internal load balancer, hacia los apps servers, con una arquitectura similar a la anterior. En los 4 grupos de EC2, tienen conectados AWS Guarduty , que es una IA para detectar posibles amenazas por medio de los monitoreso que realiza. Tanto los web y app servers estàn conectado al S3 bucket para almacenar la información que no debe perderse, ya que las Vms son descartables.

El backend compuesto por RDS , primero tiene un Elasic Cache , para brindar repuestas de los datos almacenados en cache y así acelerar las consultas hechas a la DB. Tiene HA implementado, se replica hacia el otro AZ todos los datos de la DB. Luego se hacen Snapshoots que se alacenan en el S3Bucket. Por otra parte hay DynamosDb con la misma lógica de replicas hacia el otro AZ y snapshoots al S3.

Los 2 microservicios, llegan desde la API gateway y pasan a las AWS Lambda con autoscaling y de ahí los datos son almacenados en la DynamoDB.

Tanto los 4 grupos de EC2, con los GuardDuty, y los grupos Lambda , son monitoreados por el CloudWatch, que envia esta info al AWS SNS, para enviar notificaciones a los engineer o administradores a cargo de la VPC.

Por último el S3bucket, tiene una replica en otra región como respaldo.