Prueba Tecnica DevSu

Nicolas Bendeck

GitHub> https://github.com/NicolasBendeck/3TierarchwithDR

Resumen

Arquitectura sencilla de operar, segura y que puede crecer sin tener problemas de escalabilidad. Propongo un diseño 3-tier en AWS Web - App -Base de datos con distribución global de contenido CDN, almacenamiento de objetos en S3 y una segunda región en modo warm-standby para recuperación ante desastres. Busco equilibrio entre confiabilidad y costo: alta disponibilidad en la región primaria y un plan B listo en minutos.

Proveedor elegido: AWS

Ecosistema: CloudFront, WAF, ALB, EC2, RDS, S3 y AWS Backups.

Alta disponibilidad: Multi-AZ y capacidades multi-región.

Costos controlables: auto-scaling, S3 barato para estáticos y VPC Endpoints para evitar tráfico por NAT.

EC2/RDS/S3/CloudFront reduce riesgo operativo.

Arquitectura propuesta

Route 53-CloudFront CDN con WAF adjunto.

Estáticos imágenes, JS/CSSdesde S3 con Origin Access Control .

Tráfico dinámico a un Application Load Balancer público que apunta al tier Web , subnet privada.

Del Web Server se enruta al ALB interno y de ahí al App Server.

El App Server se conecta a Amazon RDS MultiAZ para escribir datos.

Subidas de archivos van a S3 por un Gateway VPC Endpoint (sin Internet).

Backups con AWS Backup y copia cruzada a otra región. RDS mantiene una réplica de l lectura cross-region.

Componentes y su razón de ser

Frontend: CloudFront + WAF bajan latencia y filtran ataques en el borde. S3 sirve estáticos sin cargar servidores.

Backend: Separar Web y App con dos ALBs permite escalar cada capa y simplifica la escalabilidad.

Base de datos: RDS Multi-AZ evita caídas por falla de AZ y facilita backups,snapshots.

Almacenamiento de objetos: S3 para estáticos y medios, más seguro y barato.

Disponibilidad y Disaster recovery

En la región primaria todo corre en Multi-AZ. En la secundaria dejamos ASGs mínimos y una réplica de RDS.

Si la primaria falla:

1promovemos la réplica a writer,

2 apuntamos CloudFront/Route 53 al ALB de la region de disaster recovery.

Elegí warm-standby por su habilidad de estar en produccion en minutos gracias a los backups tanto de app como de DB.

Seguridad

Perímetro: WAF en CloudFront

Red: solo el ALB público expuesto; Web-App-DB en subnets privadas.

Acceso: Security Groups de mínimo privilegio (ALB-Web, Interno-App, App-RDS).

Datos: S3 con Block Public Access + SSE KMS

Escalabilidad

Horizontal en Web y App con Auto Scaling.

CloudFront cachea y reduce request al origen.

RDS escala verticalmente y con réplicas.

S3 escala prácticamente sin límite para estáticos y medias de usuario.

Costos

S3 y CloudFront para estáticos es mas barato y descarga al origen.

Auto Scaling con costo mínimos bajos con reserved instances.

Distaster recovery warm-standby: se paga poco día a día y se escala solo en el desastre.

Operación y observabilidad

CloudWatch métricas/alertas y tableros por servicio.

Despliegues rolling o blue-green.

Pruebas de failover periódicas

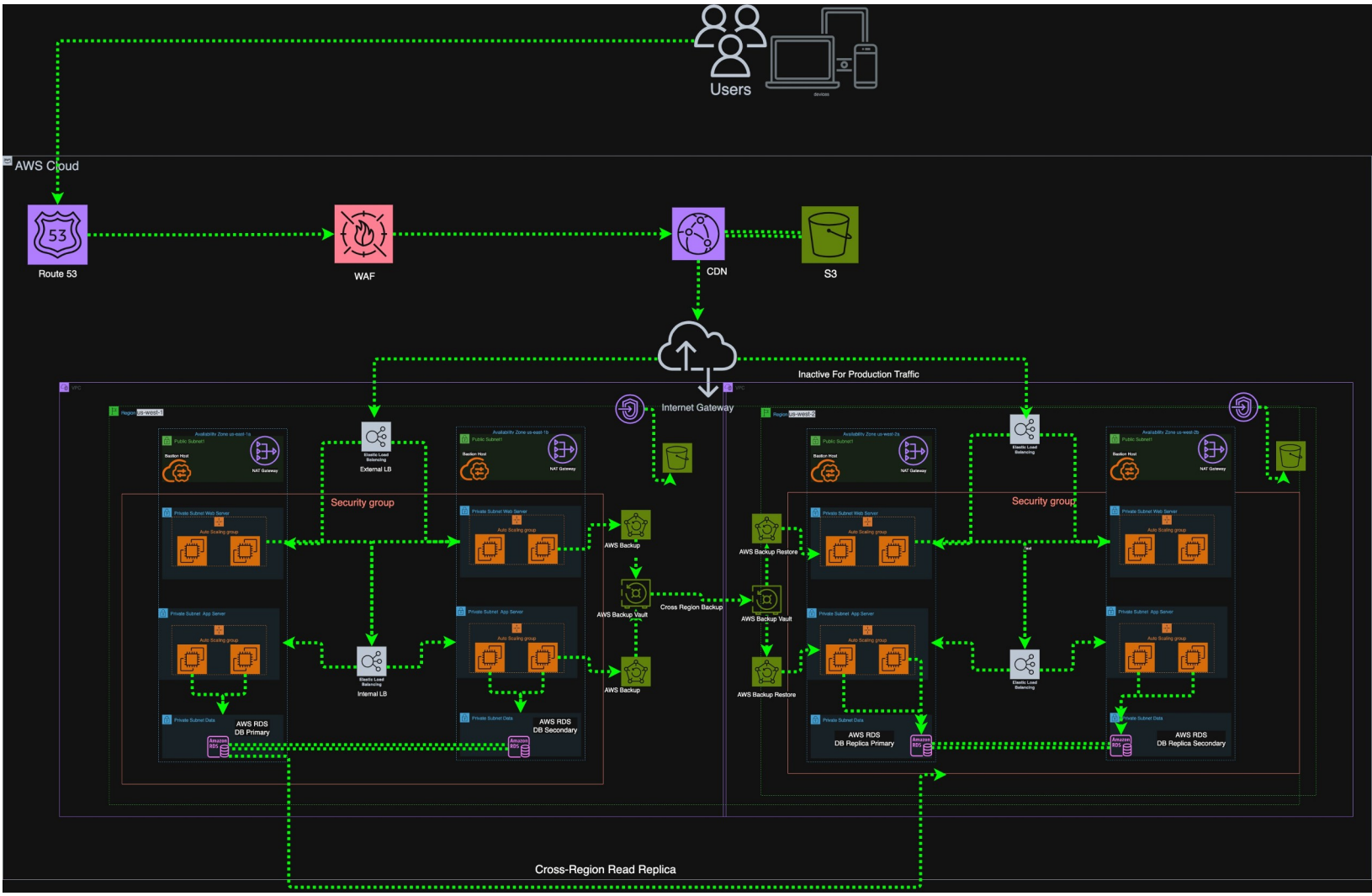
IaC Terraform o Cloud Formation.

Riesgos y mitigaciones

Tráfico pico inesperado → CDN + Auto Scaling.

Contención en DB → monitoreo de IOPS/CPU y plan de migración a Aurora/Read Replicas.Errores de configuración y revisiones de cambio, cuentas separadas por entorno y backups verificados.

Arquitectura de 3 Tiers Con Disaster Recovery.



Preguntas y Respuestas

>1. ¿Cuál es la diferencia entre nube pública, privada e híbrida?

- La nube privada> Es un data center local, que tiene sus desventajas como problemas para escalar rapidamente los recursos.

-La Nube Publica> es una infraestructura y servicios ofrecida por proveedores como AWS, Azure o GCP, esto nos permite tener una escalabilidadcasi infinita y desplegar recursos en minutos, por otra parte la reduccion de costos es significativamente reducida y cuenta con alta disponibilidad ya que cuenta con regiones presentes en ubicaciones estrategicas alrededor del mundo con escalabilidad casi infinita en minutos y hay menos precupaciones del mantenimiento de hardware y configuraciones del mismo.

La Nube Hybrida> Es una combinacion de nube publica y privada, esto la hace flexible, por ejemplo datos sensibles en la nube privada y los servicios en la nube publica, una desventaja es que la proteccion de ambos entornos ante un ataque es mas dificil de manejar.

>2. Describa tres prácticas de seguridad en la nube.

1> Monitoreo y Alertas: Es super importante estar alerta lo que pasa en los sistemas de nuestra nube, hay que vigilar constantemente la actividad como accesos, el consumo de recursos, y tener alertas automaticas. Herramientas que usaria para tener la observabilidad

necesaria serian, DataDog, Splunk y con nativas de la nube como CloudWatch etc.

>3. ¿Qué es IaC, y cuáles son sus principales beneficios? Mencione 2 herramientas de IaC y sus principales características.

La infraestructura como codigo es la forma de provisionar y gestionar infraestructura usando codigo en otras palabras tu infraestructura queda con versiones y control para los cambios, y con modulos faciles de desplegar repetitivamente sin errores humanos, lo que aumenta la productividad a la hora de hacer cambios de infraestructura sin errores que pueden costar miles de dolares en minutos.

Herramienta que yo usaria> Terraform ya que es agnostica puede usarse en cualquier cloud.

>4. ¿Qué métricas considera esenciales para el monitoreo de soluciones en la nube?

-Uso de recursos> utilizacion de CPU, memory disco, ancho de banda de red.

_Metricas de aplicacion que nos permite tener observabilidad de los recursosnos permite ver el consumo de recursos de la aplicacion, cuellos de botella, y erroes por medio de logs.

-Salud de servicios como por ejemplo ver el uptime, la latencia de respuesta y errores 4xx y 5xx, asi asegurar que la aplicacion no tenga downtime y que responda rapido y con poca latencia a los usuarios.

-Seguridad es sumamente crucial en la nube, algo que hay que vigilar de cerca, por ejemplo los intentos fallidos de login desde accesos de ubicaciones inusuales, o hasta un cambio de configuracion no autorizada una herramienta efectiva seria AWSConfig para ver el historial de configs de los recursos y hasta generar alertas si algo es configurado de forma manual.

-Costos de Operacion, en este caso es importante tambien los costos por el uso de recursos y mantener los costos lo mas optimizados posible ya que por ejemplo los recursos ociosos que funcionan sin dar ningun servicio y generan un costo, en este caso es importante llevar de cerca los costos por medio por medio de herramientas como AWS cost explorer que permite tener observabilidad de los gastos de en AWS y la cuenta vinculada, tambien permite hacer un pronostico basado en un patron de uso historico.

>5. ¿Qué es Docker y cuáles son sus componentes principales?

- Docker es la herramienta que nos permite empaquetar aplicaciones en contenedores algo que es ligero y puede correr en en cualquier sistema operativo, y los componentes de Docker son:

1-Docker Engine es el responsible de gestionar imagenes y contenedores

2-Docker Images son paquetes ejecutable que contiene las intrucciones para crear un contenedor y cuenta con todo lo necesario para ejecutar la aplicacion, como el codigo.

Una docker image puede ser utilizada en en multiples donde se definen los containers.

3-Dockerfile son es un archivo de texto que definen como sera construido el container asi como las dependencias.

4-Docker CLI es la herramienta de linea de comando para interactuar con Docker, por ejemplo: docker run, docker build, etc.

5- Docker container son las instancias en ejecucion de las imagenes.

6-Docker Hub es el repositorio donde se almacenan y comparten imagenes.