Prueba Practica Ingeniero Cloud

Diseñada por: Edwin Rubiano Carreño
email:edwin.arubiano@gmail.com
Tel: (57) 3502934279

1) **Cual es la Diferencia entre nube pública, híbrida y privada.**
   Cada modelo ofrece niveles distintos de control, seguridad, costos y flexibilidad, lo que los hace adecuados para diferentes tipos de cargas y estrategias empresariales. La nube pública destaca por su rapidez y bajo costo inicial; la nube privada por su control y alta seguridad; y la nube híbrida por combinar lo mejor de ambos para lograr un balance óptimo entre regulación, desempeño y flexibilidad. Elegir el modelo correcto depende del tamaño de la empresa, el nivel de sensibilidad de los datos y las necesidades operativas.
   El el siguiente cuadro recopilo las caracteristicas que a mi parecer son las mas distintivas de cada tipo de nube

Característica	Nube Pública	Nube Privada	Nube Híbrida
Propiedad	Infraestructura administrada por un proveedor (AWS, Azure, GCP).	Infraestructura dedicada exclusivamente a una organización.	Combinación de nube pública y privada con integración.
Costo	Bajo costo inicial, modelo pay-as-you-go.	Alto costo inicial por infraestructura dedicada.	Costo mixto: optimización entre cargas críticas y no críticas.
Escalabilidad	Muy alta, prácticamente ilimitada.	Limitada a la capacidad del hardware propio.	Alta: combina recursos internos y externos.
Seguridad	Elevada, pero compartida según modelo de responsabilidad.	Muy alta, con control total y cumplimiento estricto.	Alta, depende de integraciones y políticas consistentes.
Control	Bajo o medio; depende del proveedor.	Total control sobre hardware, red y datos.	Balanceado: control de lo crítico y flexibilidad externa.
Implementación	Muy rápida y estándar.	Más lenta; requiere despliegue y mantenimiento propio.	Compleja: requiere integración, orquestación y networking avanzado.
Casos de uso	Startups, apps web, cargas variables, análisis de datos.	Bancos, gobierno, salud, datos altamente regulados.	Empresas que deben cumplir normas pero buscan agilidad.
Ejemplos	AWS, Azure, Google Cloud.	Nube privada, VMware, OpenStack, Azure Stack, AWS Outposts.	On-premises + AWS/Azure con VPN/Direct Connect/ExpressRoute.

2)Describa 3 prácticas de la seguridad en nube.

1. Gestión de Postura de Seguridad en la Nube (CSPM)

El CSPM es una práctica clave para garantizar que toda la infraestructura desplegada en la nube —incluyendo landing pages, APIs, bases de datos o pipelines— cumpla con configuraciones seguras desde el primer momento. Un CSPM analiza continuamente tus servicios (S3, Buckets, API Gateway, Load Balancers, DNS, VMs, bases de datos, Security Groups, etc.) y detecta configuraciones riesgosas, como puertos abiertos al público, buckets sin cifrado, políticas IAM excesivas o datos accesibles desde Internet.

Además, automatiza alertas y remediaciones, integra estándares como CIS o NIST, y mantiene un inventario claro "en tiempo real" de tu postura de seguridad. Esto evita que, al montar una landing page o cualquier aplicación pública, queden brechas expuestas por error humano o mala configuración.

2. Gestión rigurosa de identidades y accesos (IAM)

Principio de menor privilegio, MFA obligatorio, rotación de claves, roles temporales y segmentación por ambientes. Limita el acceso incluso si un servicio público (como tu landing) es atacado.

3. Cifrado de datos en tránsito y en reposo

HTTPS/TLS 1.2+, cifrado con KMS , KEY VAULT, certificados válidos y protección de datos sensibles en logs, bases de datos o formularios de la landing**.**

3) Qué es la IaC? ¿Cuáles son sus principales beneficios?, menciona dos herramientas de IaC y sus principales Características.

La Infraestructura como Código (IaC) es la práctica de definir y administrar toda la infraestructura (redes, servicios, identidades, seguridad y bases de datos) mediante archivos declarativos versionados, en lugar de configuraciones manuales. Esto permite desplegar ambientes completamente reproducibles, auditables y alineados con estándares de seguridad, reduciendo el riesgo de errores humanos y drift entre entornos. La, IaC me permite automatizar despliegues, validar cambios antes de producción, integrar seguridad "shift-left" y garantizar que la infraestructura responda igual en DEV, QA y PROD. Además, facilita despliegues consistentes, rollback rápidos, control de versiones, pruebas de resiliencia y reducción del MTTR. En resumen, IaC convierte la infraestructura en un sistema confiable, gobernado por código y optimizado para disponibilidad, eficiencia y escalabilidad en la nube.

Principales Características:

1. Velocidad y escalabilidad operativa.
2. Trazabilidad, auditoría y control de versiones
3. Automatización y despliegues confiables
4. Reducción de MTTR y mayor resiliencia

Herramientas IaC:

Terraform-Terragrunt:

Terraform se encarga de:

- Crear, actualizar y destruir recursos
- Mantener el estado (terraform.tfstate)
- Ejecutar planes (terraform plan)
- Aplicar infraestructura (terraform apply)

Terragrunt agrega:

- Manejo remoto del estado más fácil
- Reutilización de módulos sin duplicación
- Organización de ambientes (dev/qa/prod)
- Reducción del código repetido
- Orden de dependencias entre módulos
- Enfoque en arquitectura multi-cuenta o multi-entorno

Pulumi

Es una herramienta de IaC que permite definir infraestructura usando lenguajes de programación como Python, TypeScript, Go o C#. A diferencia de Terraform, que usa HCL declarativo, Pulumi permite lógica avanzada, reutilización de código y abstracciones que facilitan la automatización a gran escala. Maneja estado, es multi-nube y se integra muy bien con Kubernetes y CI/CD.

- Reutilizar funciones y librerías
- Crear abstracciones avanzadas
- Escribir lógica compleja
- Integrarse fácil con repos existentes

4)¿Qué Métricas Considera esenciales para el monitoreo de soluciones en la nube?

En la implementacion de practicas SRE a nivel de aplicación se proponen mimo las GOLDEN SIGNAL

- Latency
- Traffic
- Errors
- Saturation (saturación de recursos)

Para prácticas Finos es de vital importancia tener metrics que nos controlen el gasto de la nube por consiguiente .

- Daily Burn Rate (Tasa diaria de gasto)
- Forecast (Proyección de fin de mes)
- Top Services Cost (Servicios más costosos)
- Coverage de ahorro (Savings Plans / Reserved Instances)

5)¿Que es Docker y Cuales son sus componentes principales?

Docker es una plataforma de contenedores que permite empaquetar aplicaciones junto con todas sus dependencias (bibliotecas, configuraciones y entorno de ejecución) en unidades ligeras, portátiles y consistentes llamadas contenedores.

garantiza despliegues reproducibles y simplifica la operación en múltiples entornos: desarrollo, prueba.

Docker aísla cada aplicación del host y de otros servicios sin requerir máquinas virtuales pesadas, lo que reduce consumo de recursos, acelera tiempos de entrega y permite escalar de manera horizontal sin fricción. Además, se integra con herramientas de IaC, pipelines CI/CD, orquestadores como Kubernetes y sistemas de observabilidad, facilitando la automatización, resiliencia y confiabilidad del sistema.

Componentes:

- Docker Engine
- Docker Image
- Docker Container
- Dockerfile
- Docker Registry
- Docker Networks
- Docker Volumes

6) Ejercicio Practico.

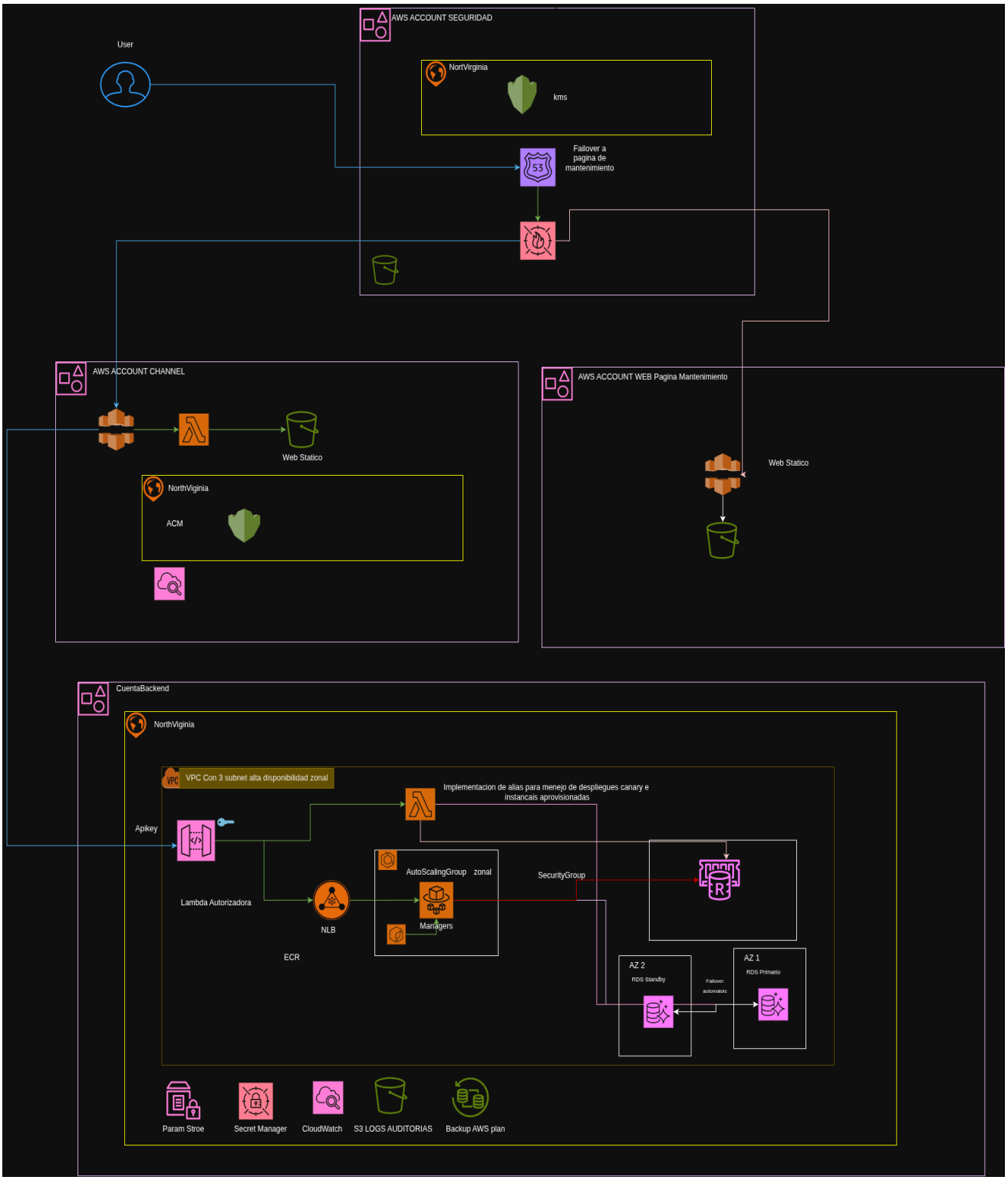
La arquitectura propuesta organiza la solución en múltiples cuentas (Canal, Backend y Seguridad), lo que permite aislar responsabilidades y aplicar controles centralizados de red, identidad y DNS. El enrutamiento externo se gestiona desde una cuenta de servicios compartidos mediante Route 53, donde se define la zona hospedada y se configuran reglas de failover hacia el punto de entrada del front. Esto asegura un punto de control único, evita configuraciones dispersas y facilita la protección del dominio.

El front puede operar en más de una región para garantizar disponibilidad del contenido, mientras que el backend se ejecuta en una región principal con despliegues automatizados, monitoreo y escalamiento según demanda. La infraestructura de cómputo está distribuida en múltiples zonas de disponibilidad mediante auto scaling y balanceo de carga, lo que permite mantener continuidad incluso si falla una AZ. La base de datos se implementa con un modelo Multi-AZ que ofrece un endpoint único y failover automático, de manera que el servicio puede recuperarse rápidamente sin cambios en el código ni intervención manual.

La solución incorpora mecanismos de cifrado, gestión centralizada de secretos, certificados y llaves KMS, así como registros de auditoría en S3 y CloudWatch para garantizar visibilidad y trazabilidad. También se contemplan alarmas, health checks, dashboards y controles de configuración para detectar desviaciones, mejorar la previsibilidad y reducir el tiempo de recuperación ante incidentes. Los recursos están diseñados para ajustarse según el uso, apoyándose en autoscaling, lifecycle policies y revisión continua de costos para asegurar eficiencia operativa y financiera.

Finalmente, toda la infraestructura puede definirse y desplegarse mediante IaC (Terraform/Terragrunt), lo que permite consistencia entre ambientes, facilidad de replicación, control de versiones y entregas más seguras. Esta combinación de separación por cuentas, resiliencia zonal, automatización, seguridad integrada, trazabilidad y eficiencia en costos demuestra que la arquitectura sigue un diseño sólido, moderno y alineado con las prácticas recomendadas por AWS para soluciones robustas en la nube.

El archivo en formato original drawio se adjunta en el repo Git y tambien se inserta la imagen en este documento


AWS ACCOUNT SEGURIDAD
User
NortVirginia
kms
Failover a pagina de mantenimiento
AWS ACCOUNT CHANNEL
Web Statico
NorthViginia
ACM
AWS ACCOUNT WEB Pagina Mantenimiento
Web Statico
CuentaBackend
NorthViginia
VPC Con 3 subnet alta disponibilidad zonal
Implementacion de alias para manejo de despliegues canary e instancais aprovisionadas
Apikey
AutoScalingGroup zonal
SecurityGroup
Lambda Autorizadora
NLB
Managers
ECR
AZ 2
RDS Standby
Failover automatic
AZ 1
RDS Primario
Param Stroe
Secret Manager
CloudWatch
S3 LOGS AUDITORIAS
Backup AWS plan