Plataforma de Sistema Bancario

Arquitectura propuesta

La presente definición de Arquitectura pretende satisfacer de acuerdo a las consignas planteadas una solución Hibrida (Cloud – On premise) para la creación de una plataforma Bancaria que ofrezca una Alta Disponibilidad y Tolerancia a fallos según los estándares actuales Internacionales de la Industria Financiera

Intoduccion

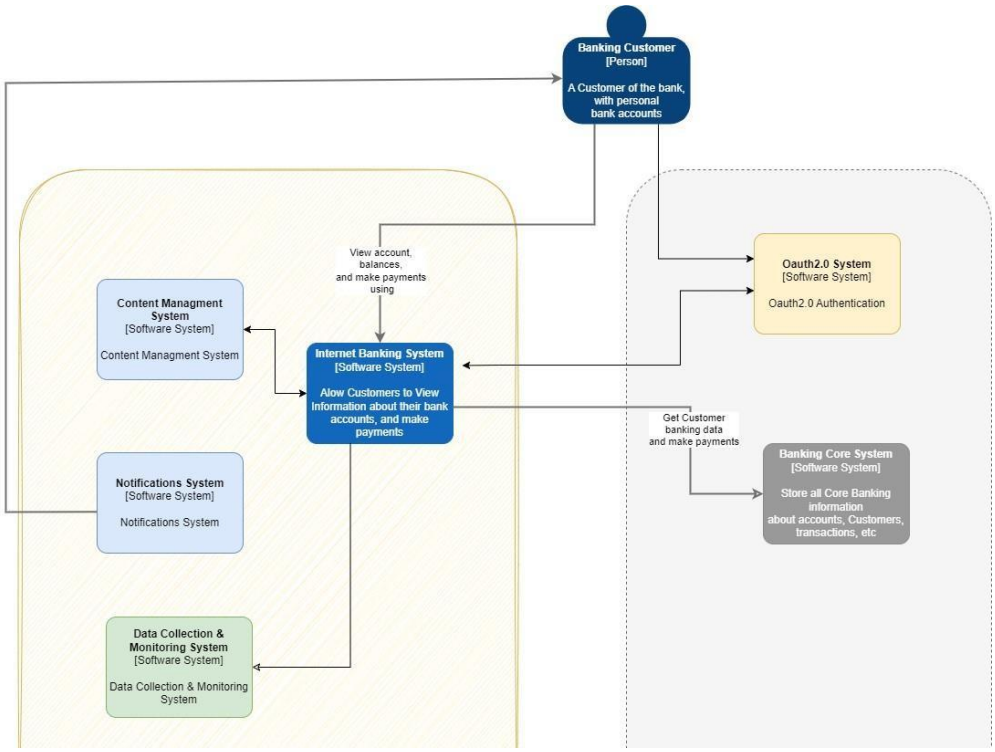
El diseño de la Arquitectura se ha desarrollado mediante el modelo C4 donde se ha abarcado según la consigna en los 3 primeros modelos :

- Diagrama de Contexto
- Diagrama de Contendedor
- Diagrama de Componentes

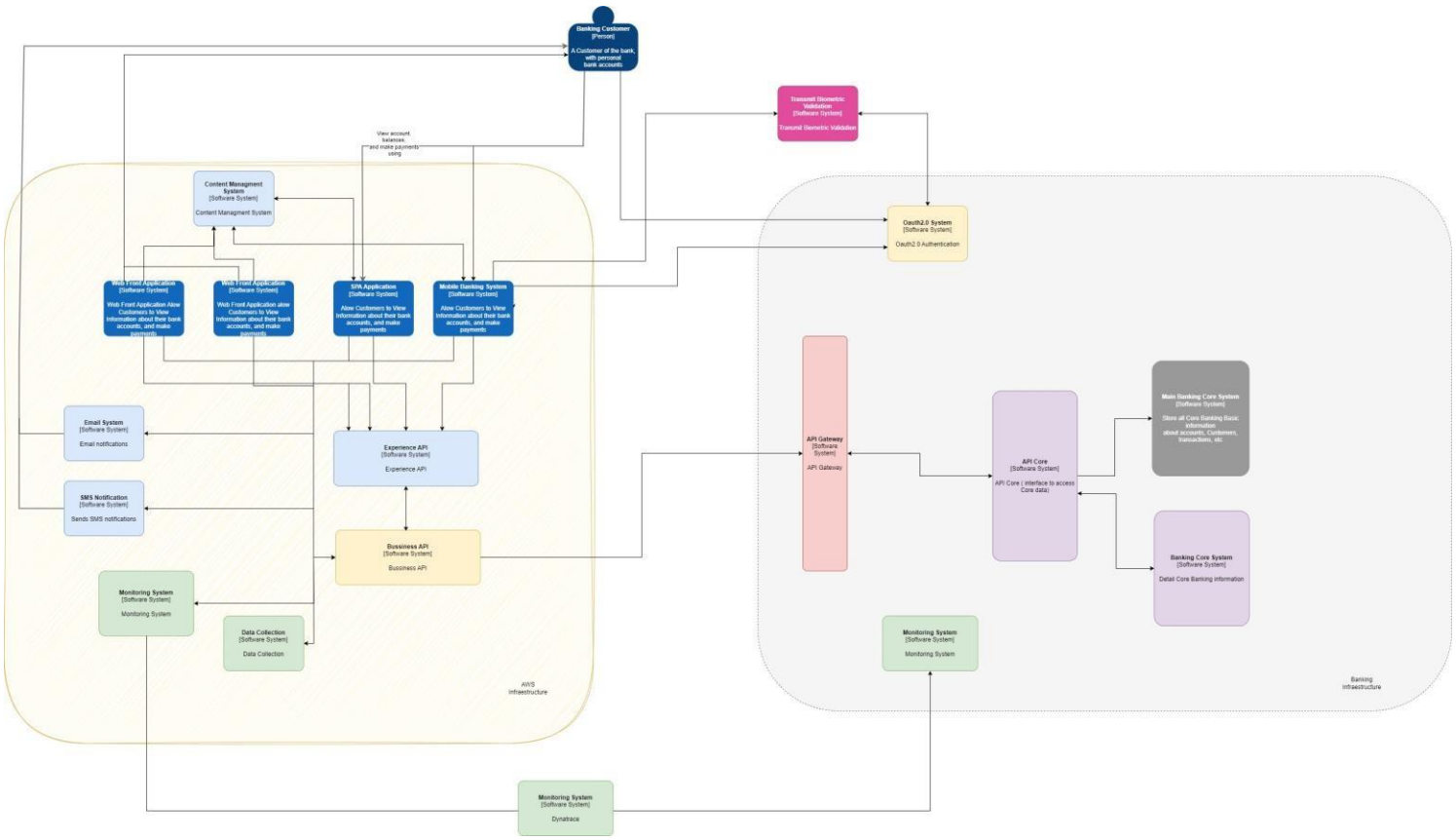
También se ha creado el diagrama de Infraestructura.
Para ello se ha utilizado Drawio como herramienta para generar los gráficos

1- Diagrama de Contexto

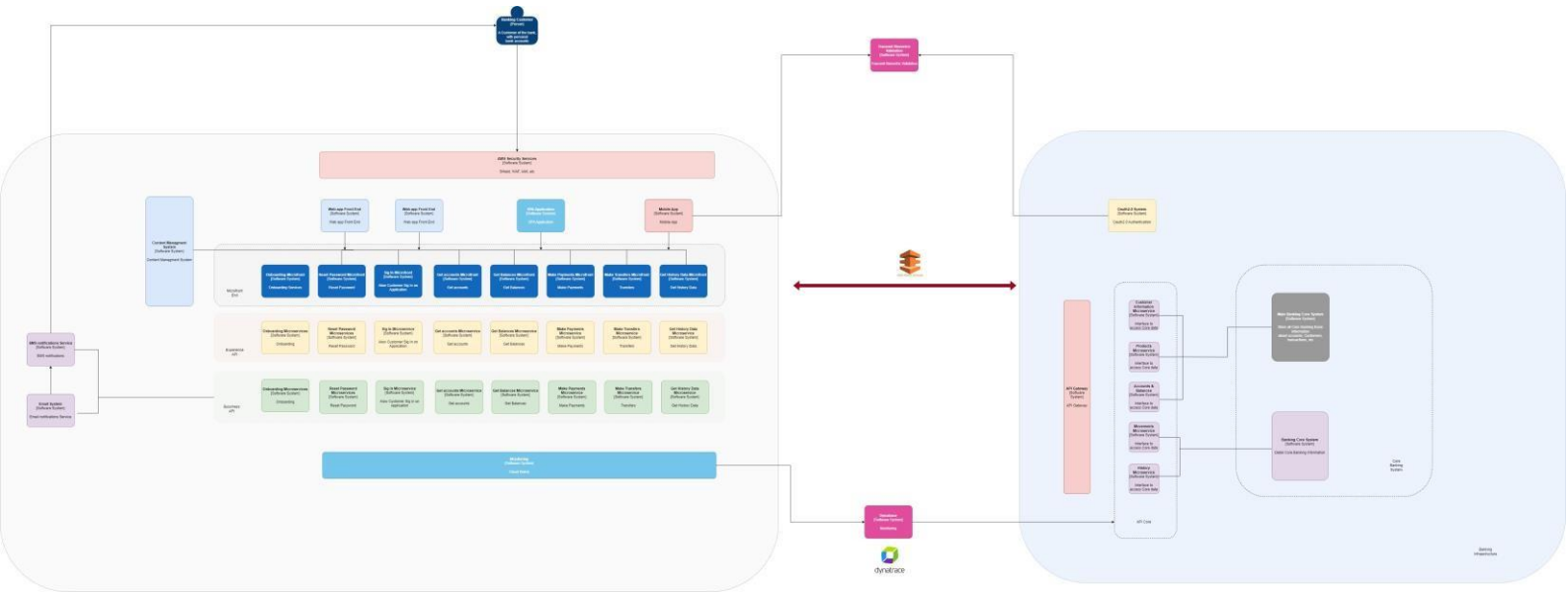
.



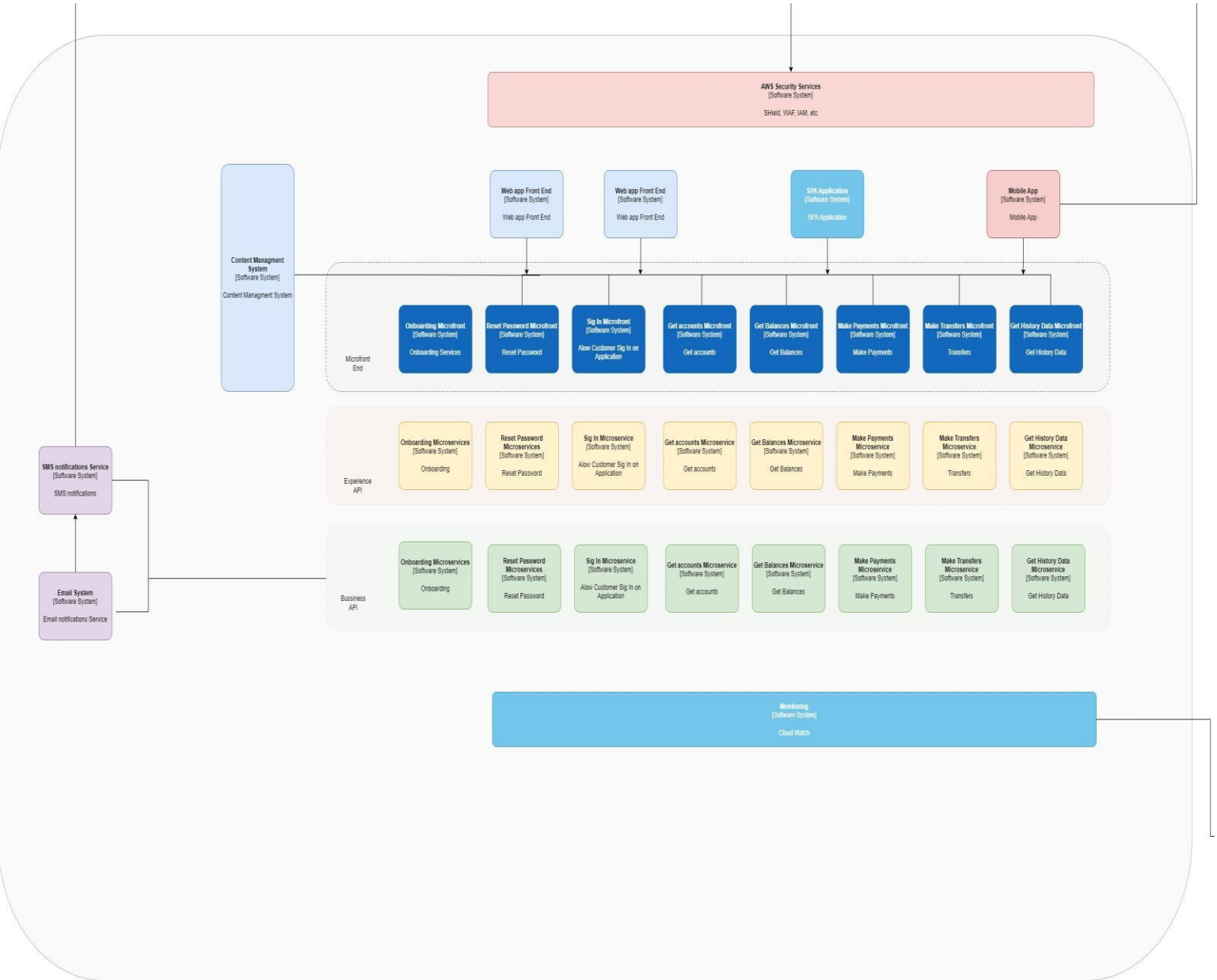
2- Diagrama de Contenedores



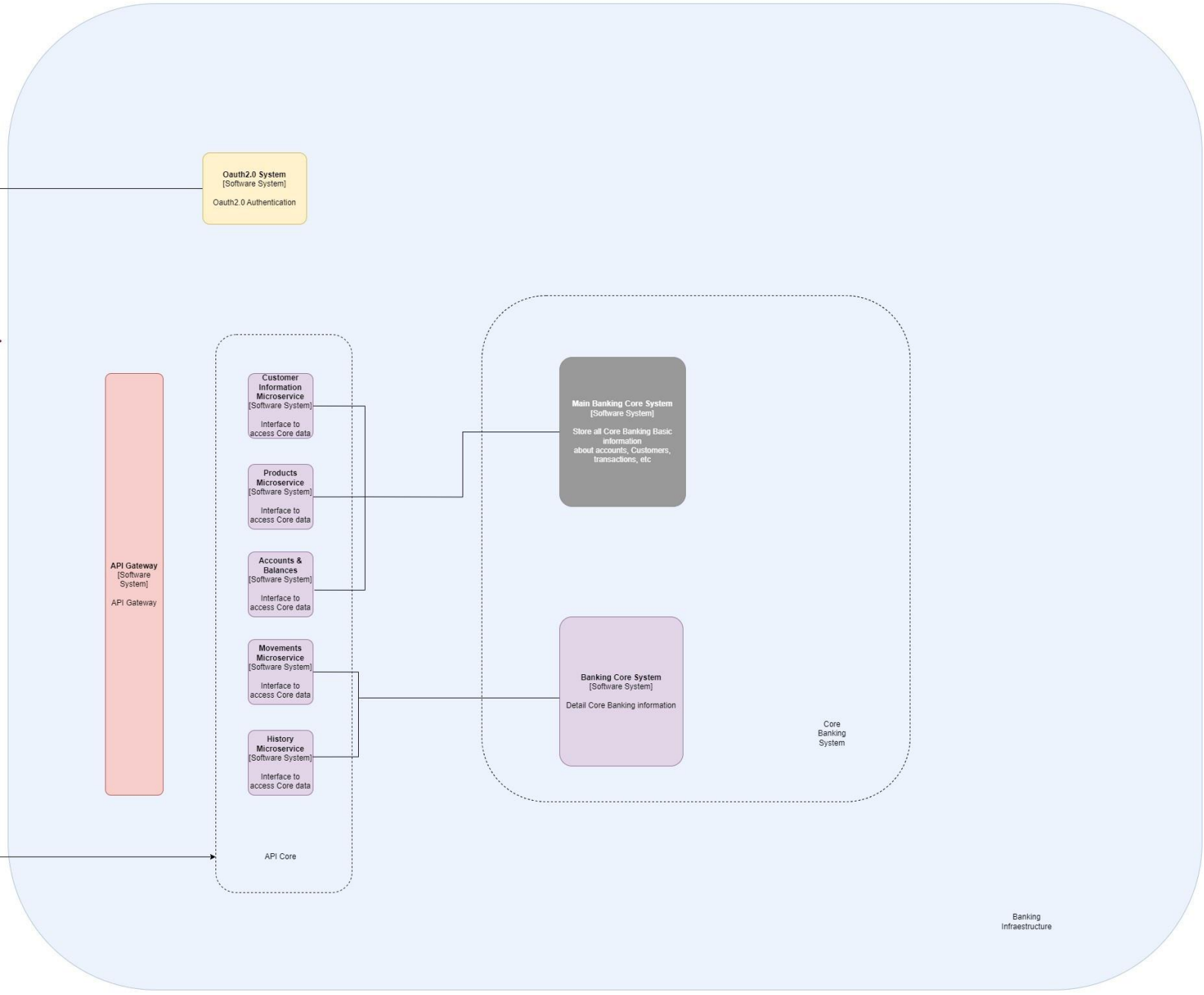
3- Diagrama de Componentes



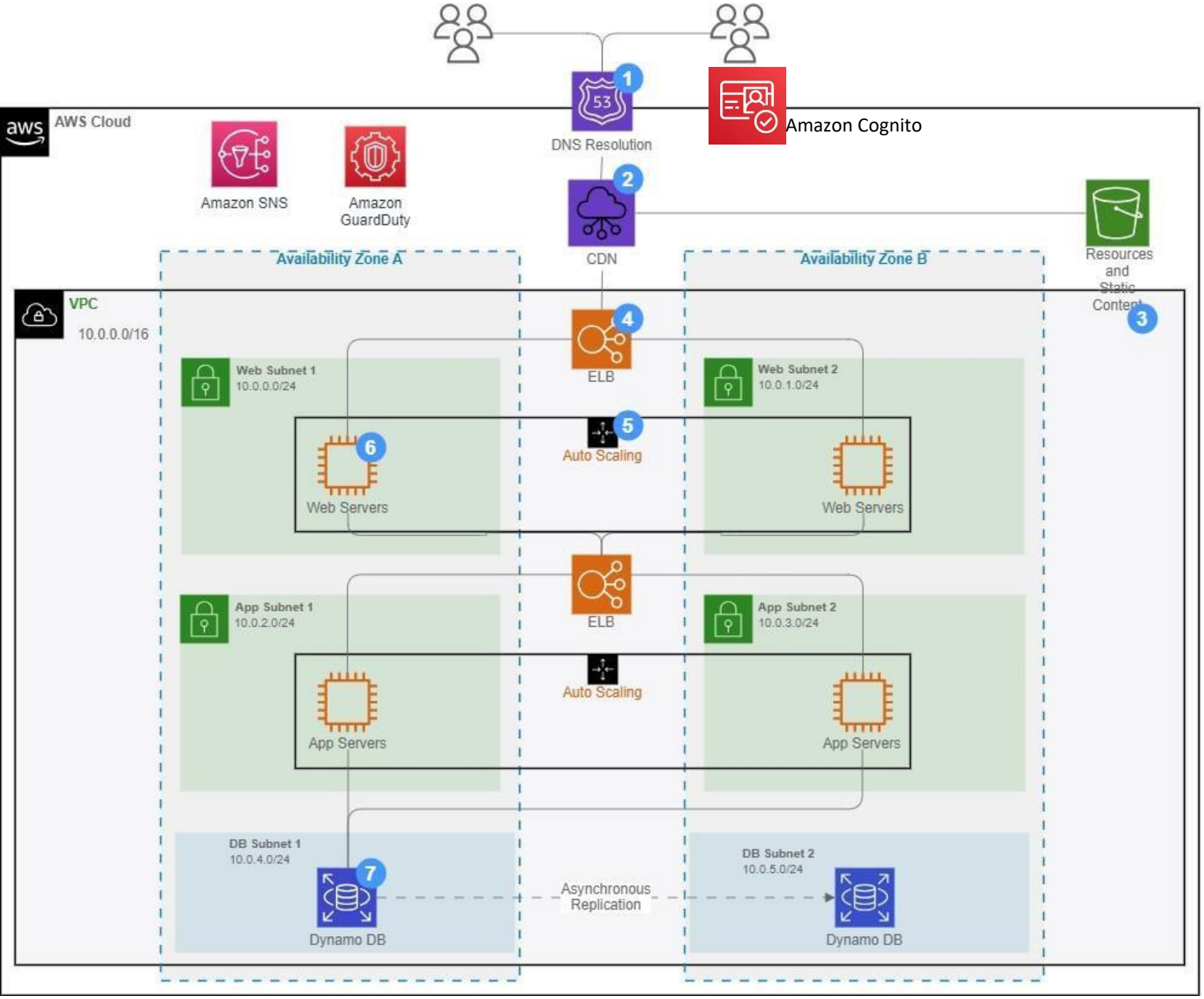
3.A- Diagrama en Aws



3.B- Diagrama en On Premise



4- Diagrama de Infraestructura



1. Las solicitudes de DNS del usuario son atendidas por Amazon Route 53, un servicio de sistema de nombres de dominio (DNS) de alta disponibilidad. El tráfico de red se dirige a la infraestructura que se ejecuta en Amazon Web Services.

2. Amazon CloudFront, una red global de ubicaciones perimetrales también conocida como red de entrega de contenido (CDN), entrega contenido estático, de streaming y dinámico. Las solicitudes se enrutan automáticamente a la ubicación perimetral más cercana, por lo que el contenido se entrega con el mejor rendimiento posible.

3. Los recursos y el contenido estático utilizados por la aplicación web se almacenan en Amazon Simple Storage Service (S3), una infraestructura de almacenamiento altamente duradera diseñada para el almacenamiento de datos primarios y de misión crítica.

4. Las solicitudes HTTP son manejadas primero por Elastic Load Balancing (ELB), que distribuye automáticamente el tráfico entrante de aplicaciones entre múltiples instancias de Amazon Elastic Compute Cloud (EC2) en zonas de disponibilidad (AZ). Permite una tolerancia a fallos aún mayor en sus aplicaciones, proporcionando sin problemas la cantidad de capacidad de equilibrio de carga necesaria en respuesta al tráfico entrante de aplicaciones.

5. Los servidores web y los servidores de aplicaciones se implementan en un grupo de Auto-Scaling. Auto Scaling ajusta automáticamente su capacidad hacia arriba o hacia abajo según las condiciones que usted defina. Con Auto Scaling, puede asegurarse de que la cantidad de instancias de Amazon EC2 que utiliza aumente sin problemas durante los picos de demanda para mantener el rendimiento y disminuya automáticamente durante la demanda para minimizar los costos.

6. Los servidores web y los servidores de aplicaciones se implementan en instancias Amazon EC2. La mayoría de las organizaciones seleccionarán una imagen de máquina de Amazon (AMI) y luego la personalizarán según sus necesidades. Esta AMI personalizada se convertirá en el punto de partida para el futuro desarrollo web.

7. Para proporcionar alta disponibilidad, la base de datos no relacional que contiene los datos de la aplicación se aloja de forma redundante en una implementación Multi-AZ (múltiples zonas de disponibilidad, aquí A y B) de Amazon Dynamo DB.

4- Consideraciones

- La arquitectura del Front End se ha diseñado con Microforntends para que puedan ser independientes, reutilizables en las Web Apps y en la SPA
- Para la App Mobile podría utilizarse algún Framework como Cordova / Ionic para reutilizar los componentes web, sin embargo, una mejor opción sería utilizar un framework multiplataforma como React Native o Flutter ya que poseen un mejor rendimiento y manejo de funciones nativas.
- La mejor opción para la App mobile es desarrollar las apps de IOS y Android Nativas, pero encarece el costo de los equipos de desarrollo.
- El Monitoreo y Auditoria se realiza mediante los servicios de AWS y Dynatrace
- Para el gobierno y estrategia de Apis y microservicios se puede utilizar el standard de BIAN y SwaggerHub.
- Puede utilizarse una Arquitectura Orientada a eventos para descoplar las funcionalidades asincrónicas con alguna Plataforma como Kafka

- Para el manejo de Biometría se utiliza los servicios de Transmit y ahorrar en tiempo de desarrollo
  https://transmitsecurity.com/

5 – Plan de Migración Gradual

1. Evaluación Inicial y Estrategia de Migración

- **Evaluación del entorno actual:** Realiza un análisis detallado del sistema on-premise, identificando componentes críticos, dependencias, y posibles cuellos de botella.
- **Definición de la estrategia de migración:** Decide entre diferentes enfoques de migración (lift-and-shift, re-platforming, re-factoring) según la criticidad de las aplicaciones y su compatibilidad con la nube.
- **Definición de objetivos:** Establece objetivos claros, como reducción de costos, mejora de la escalabilidad o aumento de la agilidad.

2. Segmentación por Fases

- **Fase 1: Migración de Servicios No Críticos**
  - **Servicios auxiliares:** Comienza con la migración de servicios no críticos, como aplicaciones internas de soporte, entornos de desarrollo y pruebas.
  - **Automatización y monitoreo:** Implementa herramientas de automatización para la migración (e.g., AWS CloudFormation, Terraform) y configura un monitoreo exhaustivo.
  - **Pruebas y validación:** Realiza pruebas exhaustivas en la nube para validar el comportamiento de los servicios.
- **Fase 2: Integración y Comunicación**
  - **Migración de APIs y Middleware:** Migra las capas de integración y comunicación, como APIs, middleware, y microservicios. Configura redes híbridas utilizando AWS Direct Connect o VPN para garantizar una conectividad segura y de baja latencia entre la nube y el core on-premise.
  - **Sincronización de datos:** Implementa soluciones de sincronización de datos entre on-premise y la nube, como bases de datos replicadas o almacenamiento compartido (e.g., AWS Database Migration Service).
  - **Balanceo de carga:** Configura balanceadores de carga para distribuir el tráfico entre las instancias on-premise y en la nube, garantizando una transición fluida.
- **Fase 3: Migración de Servicios Críticos**
  - **Core Bancario y Procesos Críticos:** Una vez que las capas de integración estén probadas, comienza con la migración de componentes críticos del core bancario, priorizando aquellos que se beneficiarían más de la nube en términos de elasticidad y resiliencia.
  - **Implementación de sistemas de respaldo:** Configura sistemas de respaldo en la nube para los datos críticos del core bancario, asegurando que haya planes de recuperación ante desastres.
  - **Pruebas de rendimiento:** Realiza pruebas de carga y estrés para asegurar que el sistema en la nube puede manejar el volumen de transacciones requerido.

5 - Plan de Migración Gradual

- **Fase 4: Optimización y Desmantelamiento On-Premise**
  - **Optimización de costos:** Revisa y ajusta los recursos en la nube para optimizar costos. Utiliza instancias reservadas, spot instances y herramientas de optimización como AWS Cost Explorer.
  - **Desmantelamiento gradual:** Desmantela gradualmente los componentes on-premise que ya no son necesarios, asegurando que se mantengan redundancias y respaldos hasta la completa estabilización.
  - **Entrenamiento del equipo:** Capacita al equipo para que adopte las nuevas herramientas y procesos en la nube.

3. Gestión de la Seguridad y Cumplimiento

- **Implementación de políticas de seguridad:** Asegura que las políticas de seguridad on-premise se reflejen en la nube. Configura IAM roles y políticas, cifrado de datos en tránsito y en reposo, y herramientas de monitoreo de seguridad como AWS GuardDuty.
- **Cumplimiento normativo:** Verifica que la migración cumpla con las regulaciones del sector bancario, como PCI DSS, GDPR u otras normativas locales.

4. Monitoreo Continuo y Mejora

- **Monitoreo y alertas:** Implementa herramientas de monitoreo como CloudWatch para asegurar la visibilidad continua de todos los sistemas, tanto on-premise como en la nube.
- **Mejora continua:** Ajusta la arquitectura conforme se identifiquen nuevas oportunidades de mejora, y mantén revisiones periódicas para asegurar que la solución sigue alineada con los objetivos de negocio.

5. Plan de Contingencia

- **Planes de recuperación ante fallos:** Define procedimientos de recuperación en caso de fallos durante la migración. Implementa mecanismos de rollback y asegura redundancias durante el proceso.

6. Comunicaciones y Gestión del Cambio

- **Plan de comunicaciones:** Mantén una comunicación fluida con todas las partes interesadas, incluyendo usuarios finales, para que estén informados sobre el progreso y posibles interrupciones.
- **Gestión del cambio:** Desarrolla un plan de gestión del cambio para ayudar a los empleados a adaptarse a la nueva arquitectura.

7- Entregable

- Repositorio GitHub :
  https://github.com/pablomacchia/devsu-test

Pablo Macchia pablo.macchia@hotmail.com