



Plataforma de Sistema Bancario

Arquitectura propuesta

La presente definición de Arquitectura pretende satisfacer de acuerdo a las consignas planteadas una solución Híbrida (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

Introduccion

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

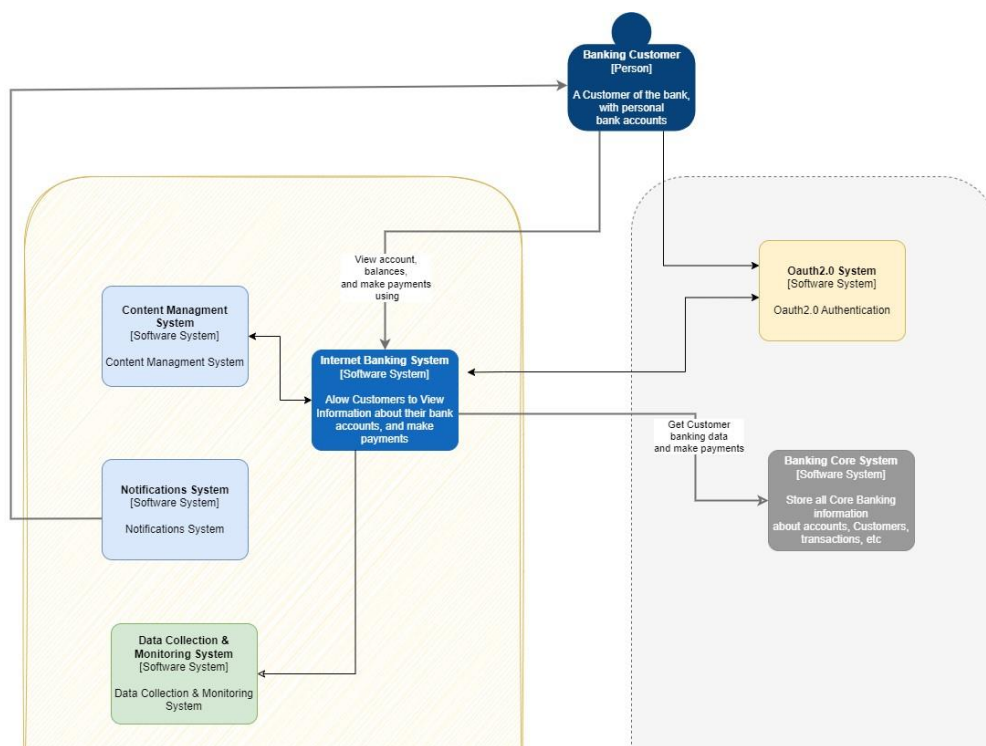
- Diagrama de Contexto
- Diagrama de Contenedor
- Diagrama de Componentes

También se ha creado el diagrama de Infraestructura.

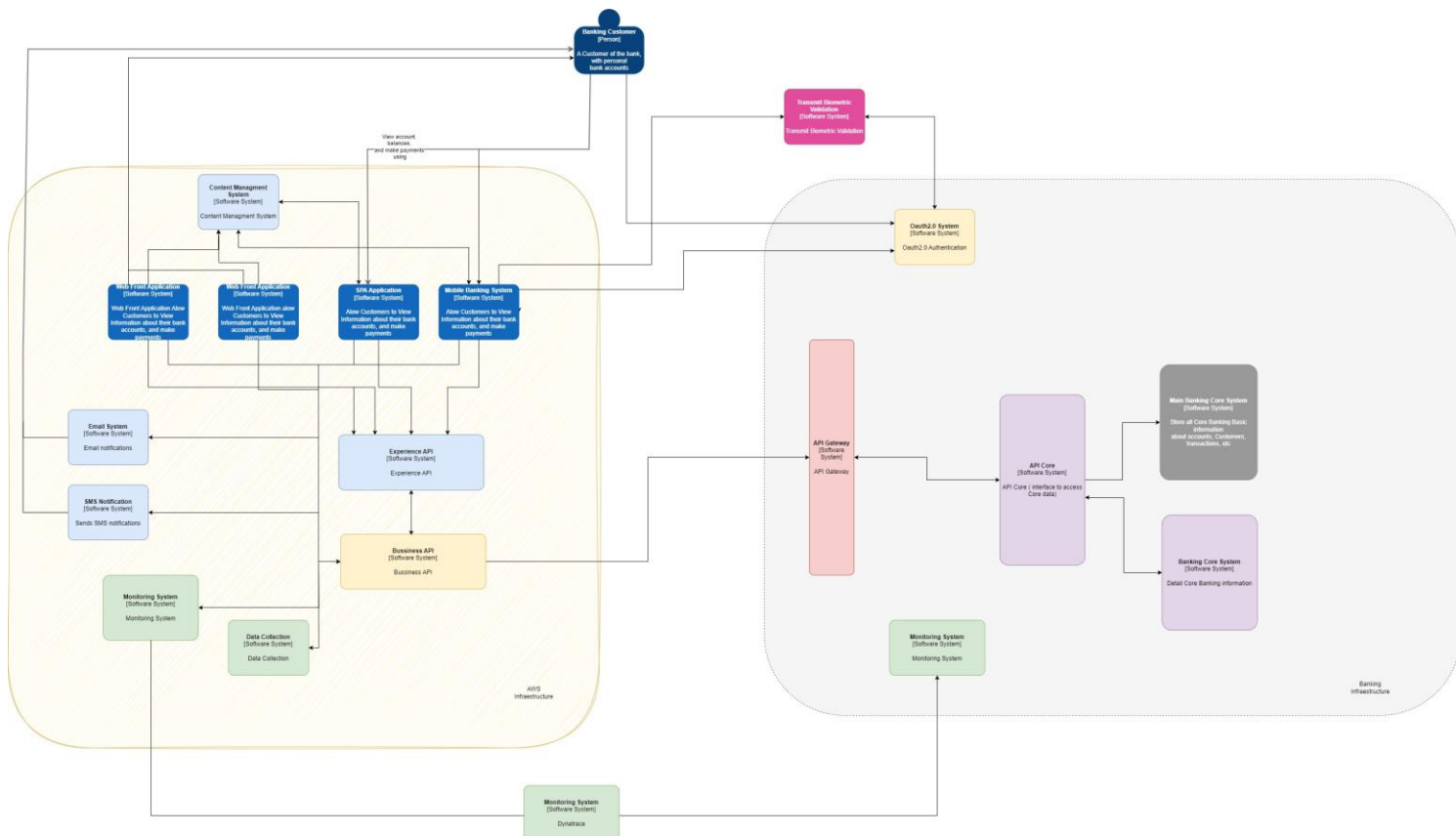
Para ello se ha utilizado Drawio como herramienta para generar los gráficos



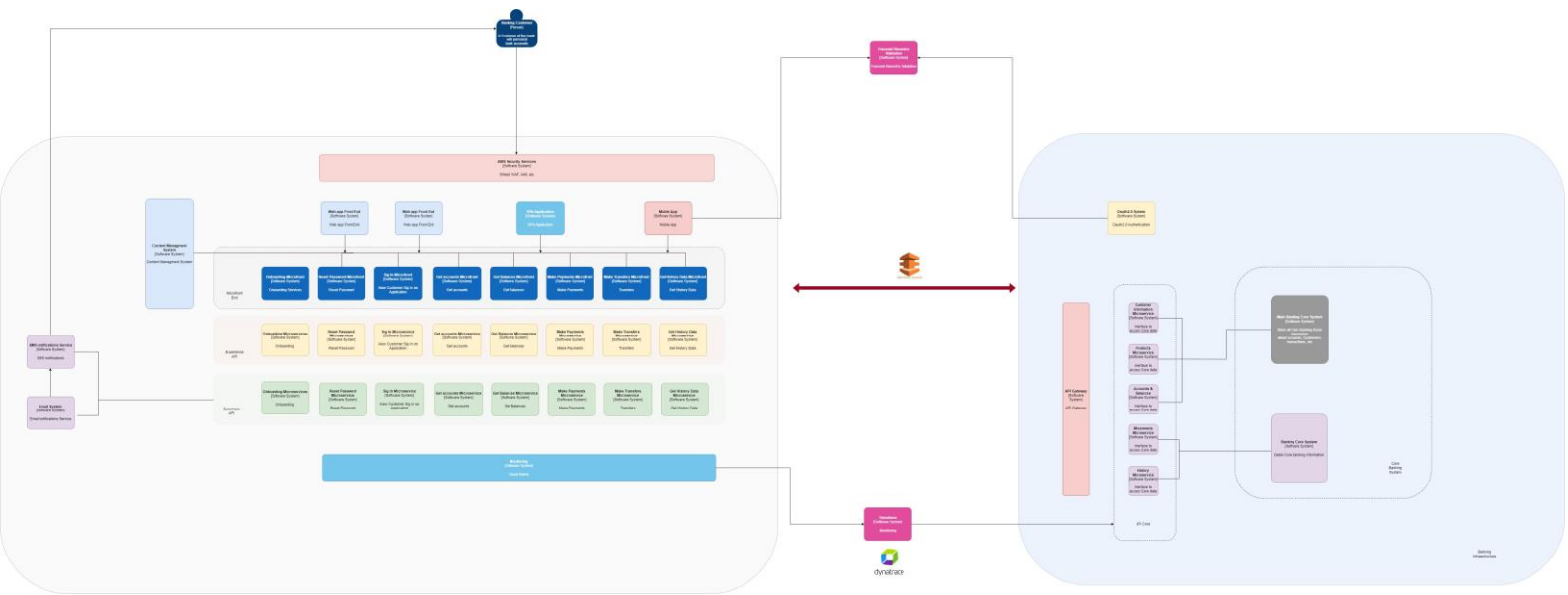
1- Diagrama de Contexto



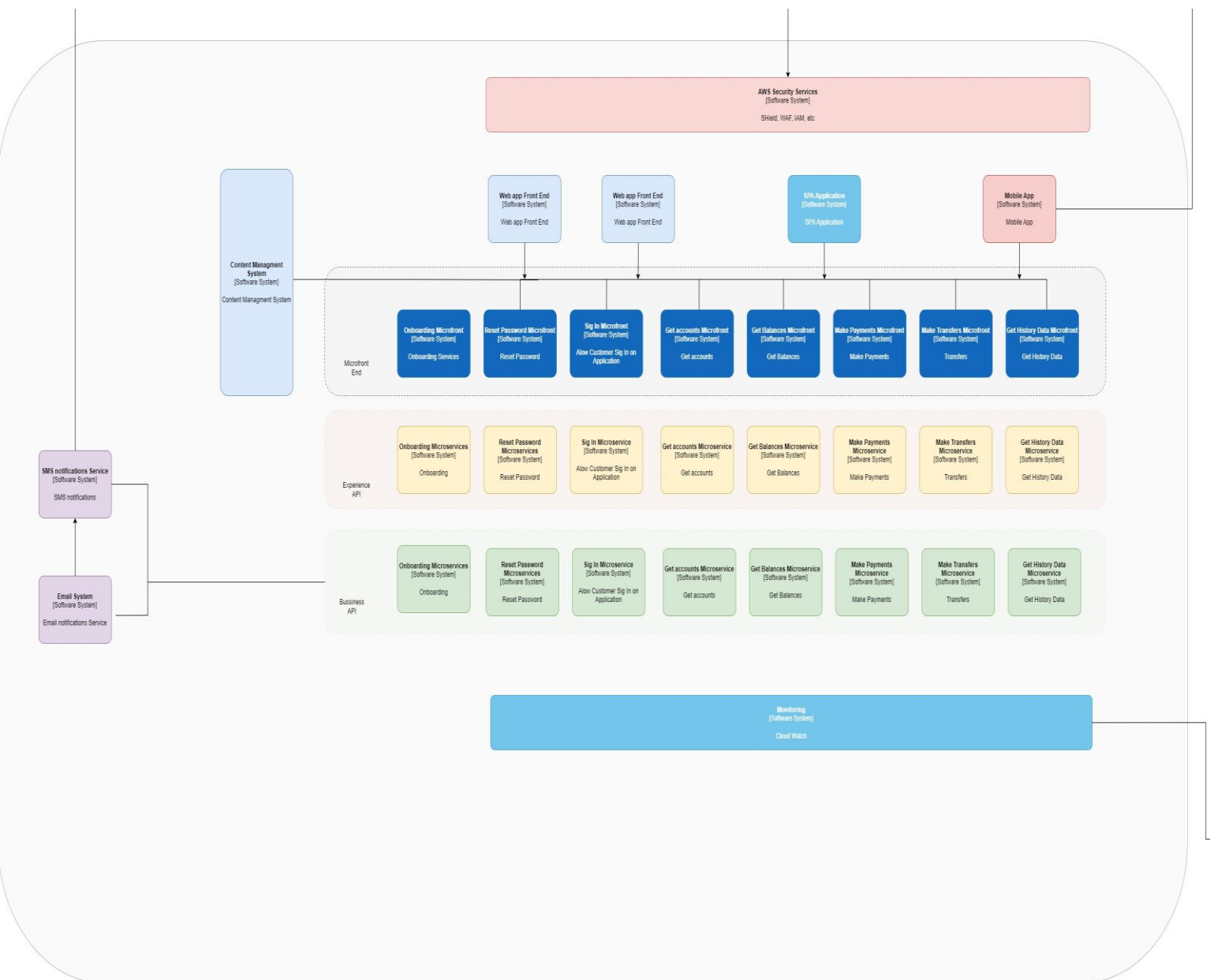
2- Diagrama de Contenedores



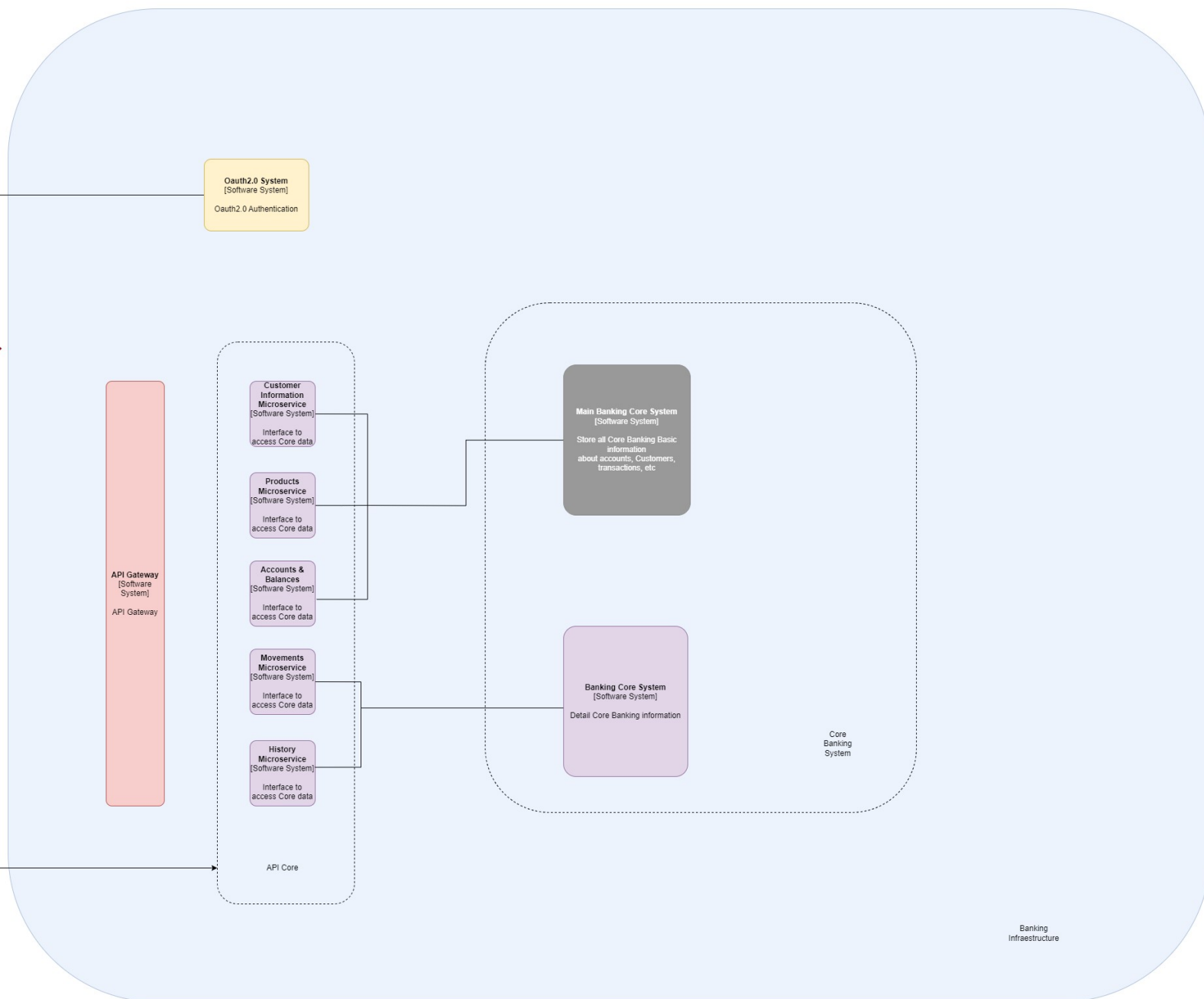
3- Diagrama de Componentes



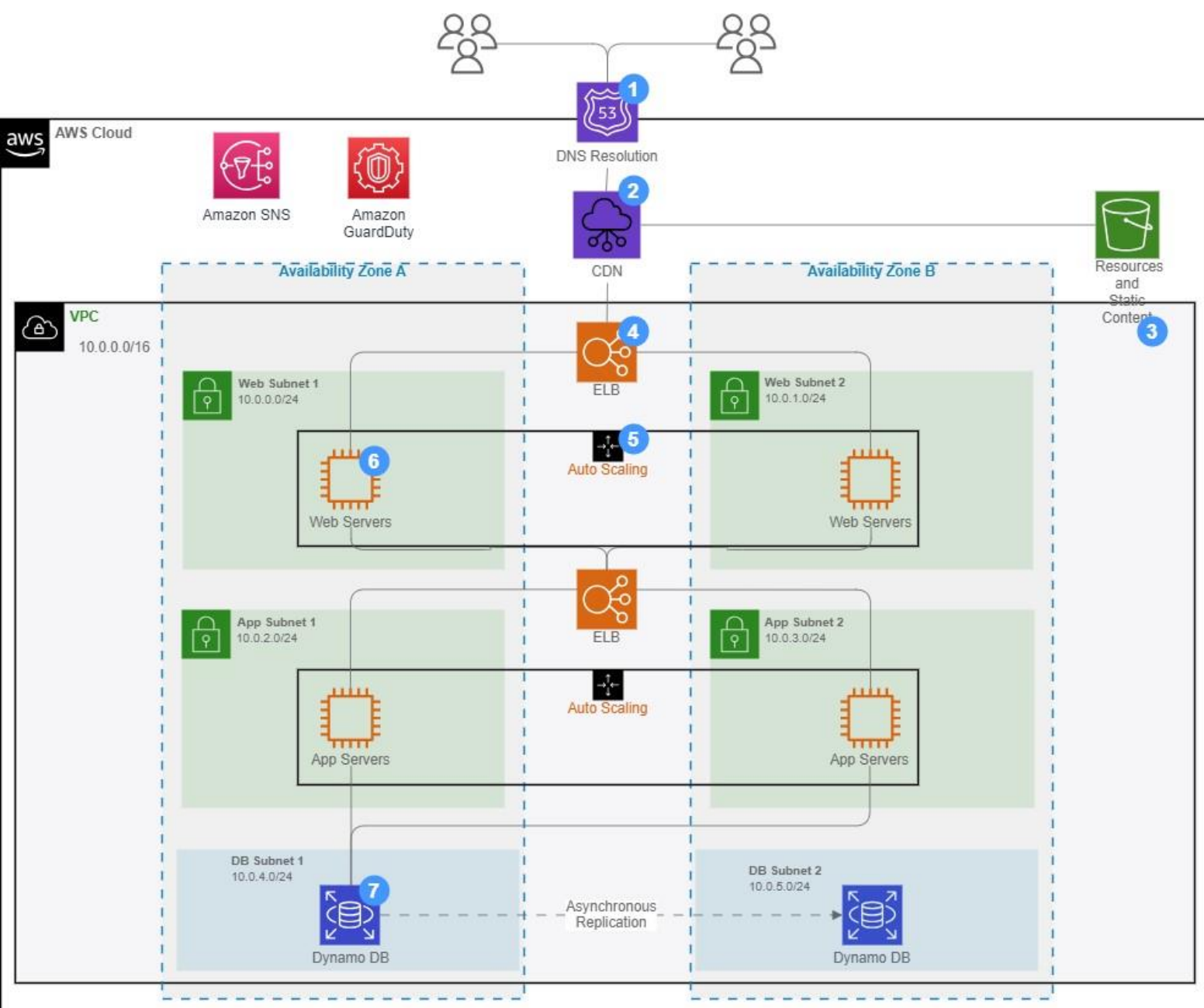
3.A- Diagrama en Aws Infraestructura



3.B- Diagrama en On Premise Infraestructura



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 manejo de Biometría se utiliza los servicios de Transmit y ahorrar en tiempo de desarrollo
<https://transmitsecurity.com/>
- Repositorio GitHub :
<https://github.com/Pablolab1/DEVSU-Test.git>

Pablo Macchia pablo.macchia@hotmail.com

