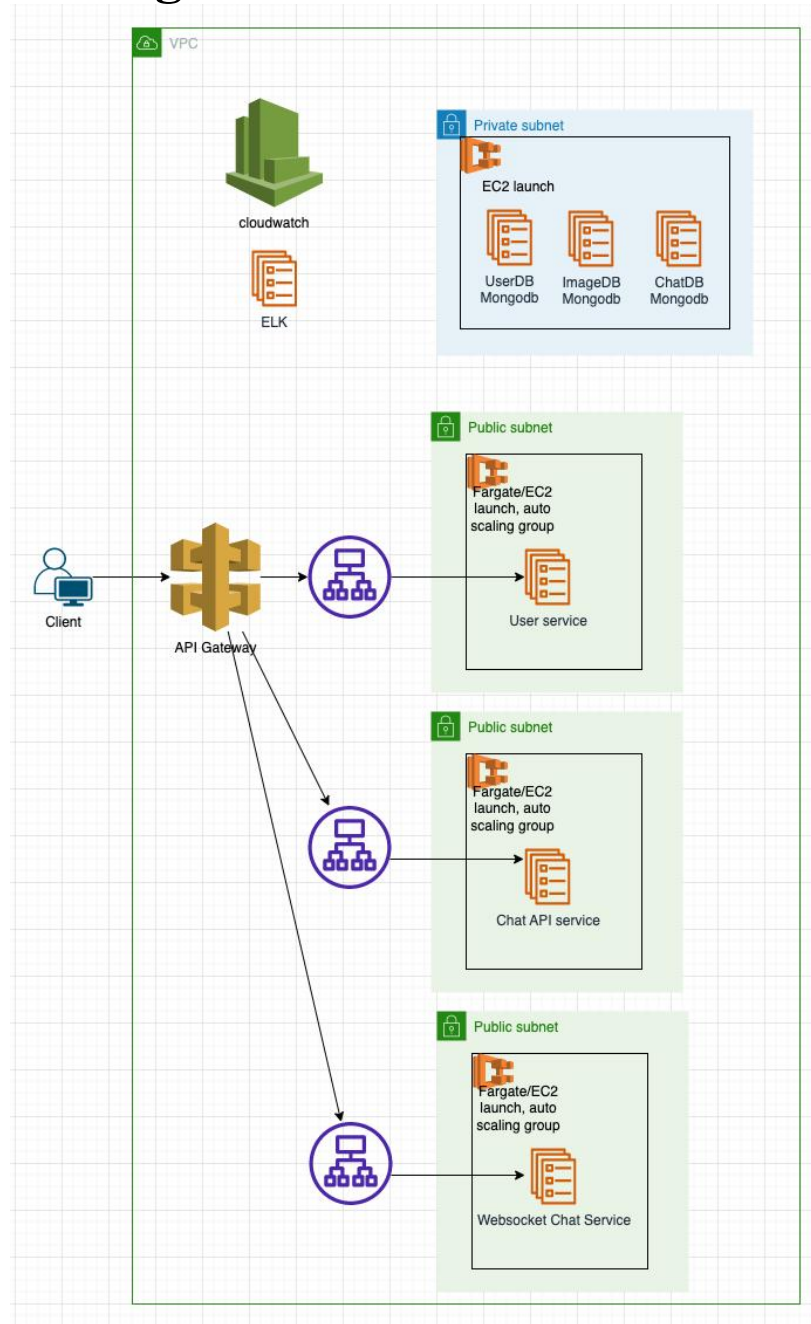


System Design

1. System Design



I will design this system base on dockersize. So the services can host on any cloud support docker (ECS, EKS, K8s, ...) or self host.

Regarding user service, used for user to register/login/auth/logout.

Regarding Chat API service, Image/Document and its data (document url, coordinate of anotations) will be store in ImageDB.

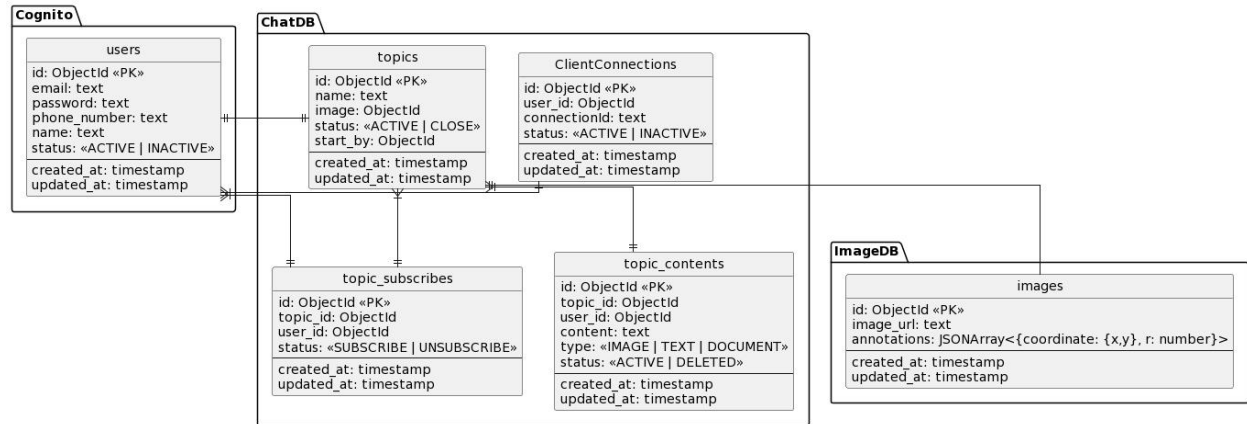
Regarding Websocket Chat Service will be use for managed connection and realtime comunications. If client of the topic doesn't connected so the message will be send by any services.(SNS, APNS, FCM, PushNub, Onesignal, ...).

For systems logs i will use cloudwatch and ELK for monitoring.

Techstack:

- Typescript.
- NestJS: <https://nestjs.com/>
- Docker.
- MongoDB.
- SocketIO: <https://socket.io/>
- ELK: <https://www.elastic.co/what-is/elk-stack>

2. Database Design



3. Resources Plan

CADDI RESOURCE PLAN						
	Feature	Design	Dev	Test	Total	Description
1	Setup Infra	1	3	1	5	Dev, Staging, Production
2	Setup code base	1	2	0	3	
3	User service	1.5	5	2.5	9	5 flows: authen, signup, login, auth, logout
4	Chat api service	2	8	4	14	create topic, get topic, delete topic, upload image, subscribe topic, unsubscribe topic.
5	Websocket chat service	2	8	4	14	connect, disconnect, send message, load message.
6	Integration test	0	0	3	3	
	Total	7.5	26	14.5	48	
Unit: Man-day						
QC:	14.5					
Dev:	33.5					