✦ NAMATUBE

مستند دیاگرام‌های معماری و طراحی سیستم – نماتیوب (Namatube)

تهیه‌کنندگان: محمدامین حاجی علیرضایی، فرازین فرهمند

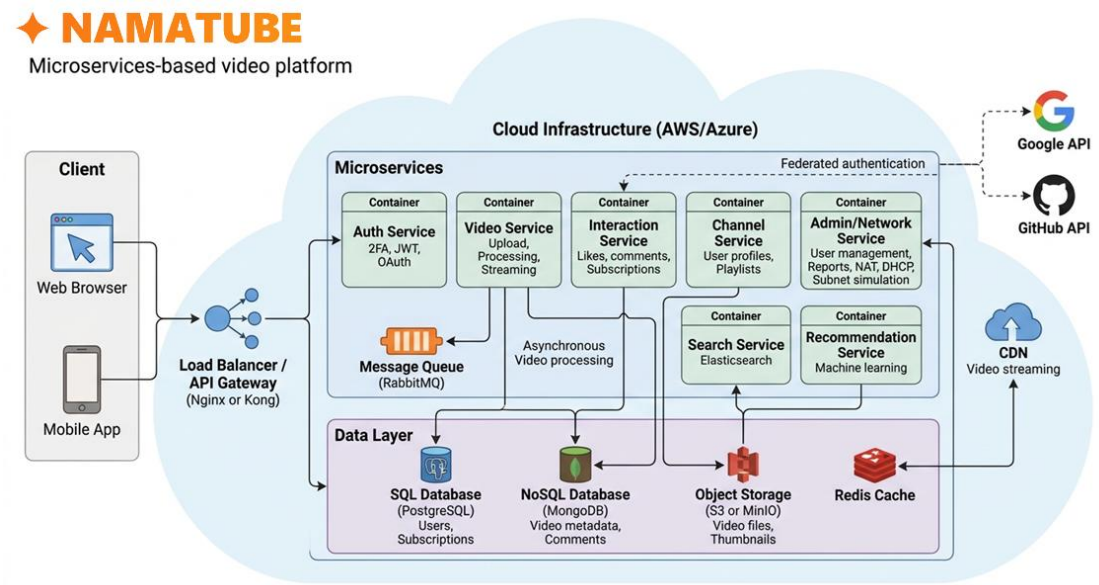
درس تحلیل و طراحی سیستم ها – دکتر صدیقیان

مقدمه

این سند شامل دیاگرام‌های کلیدی معماری و طراحی سیستم نماتیوب (پلتفرم اشتراک ویدیو) است.

1. دیاگرام معماری (Architecture Diagram)

هدف: نمایش لایه‌های اصلی سیستم، مؤلفه‌های سرویس‌ها، دیتابیس‌ها، و تعاملات خارجی.

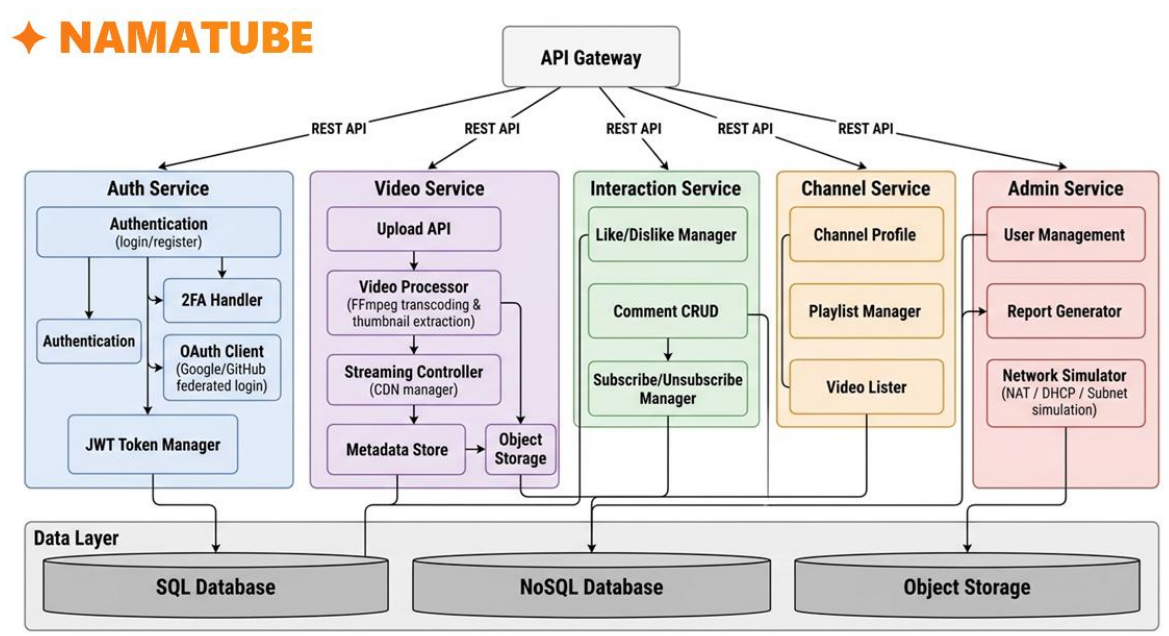


توضیح:

معماری Namatube به صورت میکروسرویس (Microservices) طراحی شده است. سرویس‌های مستقل (Auth, Video, Interaction, Channel, Admin/Network, Search, Recommendation) هرکدام در کانتینر داکر قرار گرفته و توسط API Gateway (مثلاً Nginx یا Kong) در معرض client (مرورگر/موبایل) قرار می‌گیرند. ارتباط بین سرویس‌ها از طریق REST API (با فرمت JSON) انجام می‌شود. برخی عملیات سنگین (مانند پردازش ویدیو پس از آپلود) به صورت ناهمگام با استفاده از Message Queue (RabbitMQ) انجام می‌شود. ذخیره‌سازی داده‌ها: SQL (PostgreSQL) برای اطلاعات کاربران، اشتراک‌ها و تراکنش‌ها؛ NoSQL (MongoDB) برای متادیتای ویدیوها، کامنت‌ها و برچسب‌ها؛ Object Storage (مانند MinIO یا S3) برای فایل‌های ویدیو و تصاویر بندانگشتی. همچنین برای مقیاس‌پذیری از Load Balancer و CDN برای پخش ویدیو استفاده می‌شود. سرویس ادمین شامل ماژول شبیه‌سازی شبکه (NAT, DHCP, Subnetting) است.

2. دیاگرام مؤلفه‌ها (Component Diagram)

هدف: نمایش اجزای داخلی هر سرویس و ارتباط بین آن‌ها در سطح مؤلفه (component).



توضیح:

سیستم به چندین سرویس مستقل تقسیم شده که هرکدام دارای مؤلفه‌های خاص خود است. برای مثال:

- Auth Service شامل مؤلفه‌های: Authentication (ورود/ثبت‌نام)، TwoFactorAuth، FederatedLogin (Google/GitHub)، TokenManager (JWT).

- Video Service شامل: UploadHandler، VideoProcessor (ترانسکدینگ، استخراج تامب‌نیل)، StreamManager (CDN)، MetadataManager.

- Interaction Service شامل: LikeManager، CommentManager، SubscribeManager.

- Channel Service شامل: ChannelProfile، PlaylistManager، VideoList.

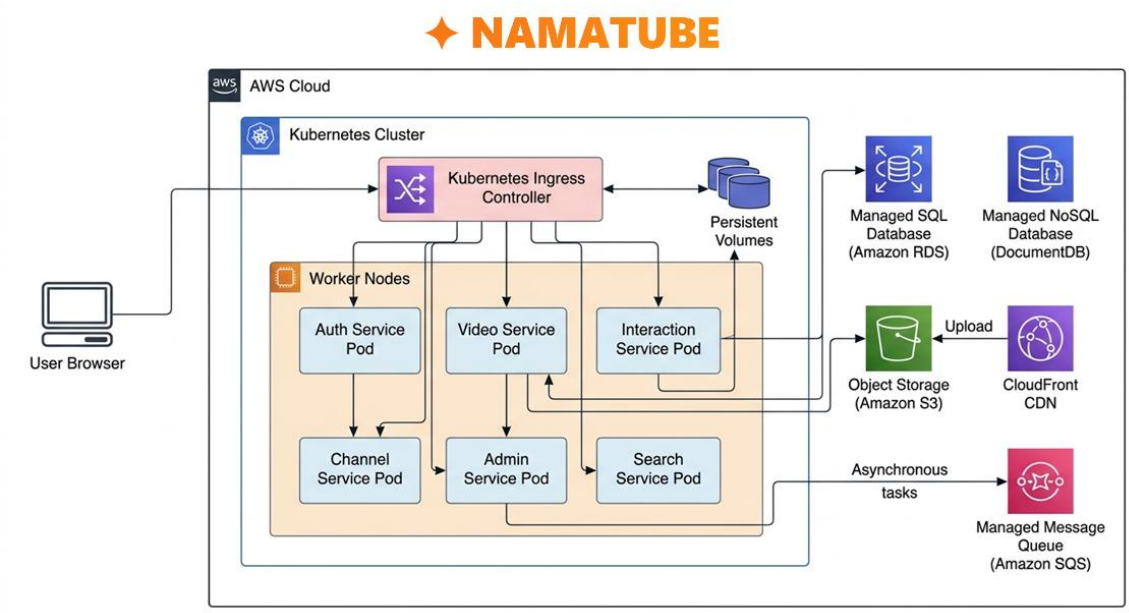
- Admin/Network Service شامل: UserAdmin، ReportGenerator، NetworkSimulator (NAT/DHCP/Subnet).

- Common Components مانند API Gateway، Message Queue، و Database Access Objects.

ارتباطات بین مؤلفه‌ها از طریق interfaces (REST API) انجام می‌شود.

3. دیاگرام استقرار (Deployment Diagram)

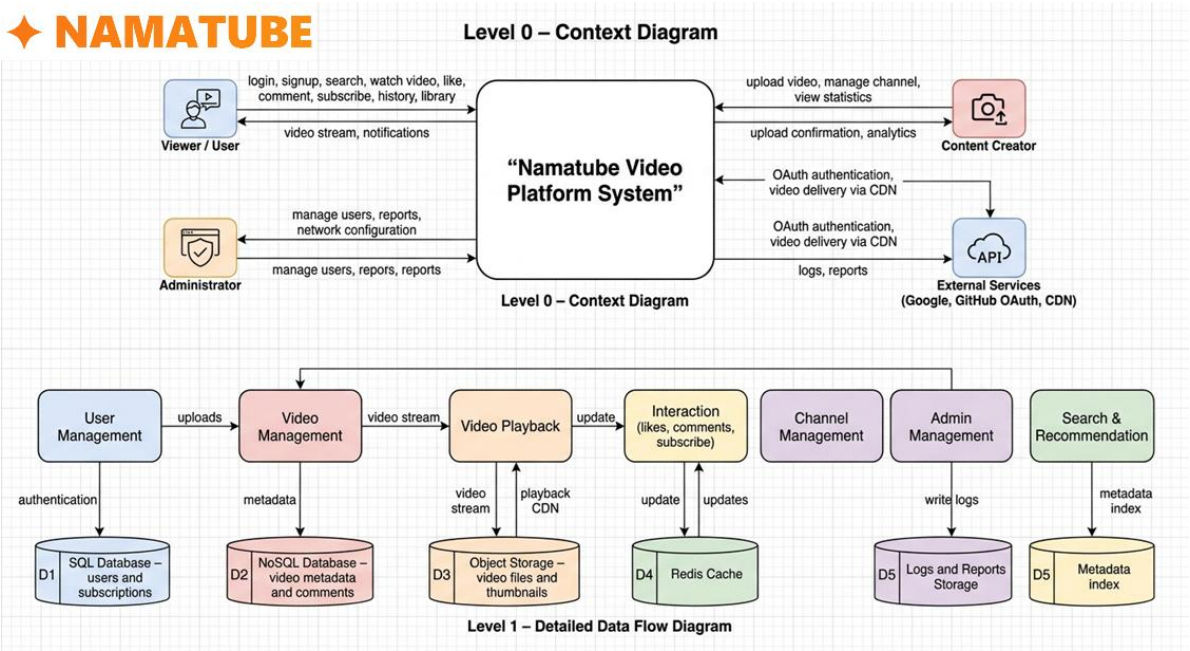
هدف: نشان دادن توزیع فیزیکی و مجازی مؤلفه‌ها روی سرورها، کانتینرها و محیط ابری.



توضیح:

پروژه Namatube بر روی فضای ابری (مثلاً AWS) مستقر می‌شود. از Docker برای کانتینری کردن هر سرویس و از Kubernetes برای ارکستراسیون استفاده می‌شود. یک Cluster Kubernetes شامل nodeهای اصلی (master) و worker. هر سرویس به عنوان یک Deployment با چندین Pod اجرا می‌شود. داده‌ها در Persistent Volumes ذخیره می‌شوند. یک Ingress Controller به عنوان Load Balancer عمل می‌کند. برای ویدیوها از CDN و Object Storage خارج از کلاستر استفاده می‌شود.

4. دیاگرام جریان داده (Data Flow Diagram)



سطح 0 (Context Diagram)

هدف: نمایش سیستم به عنوان یک جعبه سیاه و تعاملات آن با موجودیت‌های خارجی (بازیگران).

موجودیت‌های خارجی:

- کاربر عادی (Viewer)

- تولیدکننده محتوا (Creator)

- ادمین (Admin)

- سرویس‌های خارجی (Google/GitHub API, CDN)

جریان‌های داده سطح 0:

- کاربر عادی: ورود/ثبت‌نام، جستجو، تماشا، لایک/کامنت، اشتراک، مشاهده تاریخچه/کتابخانه.

- تولیدکننده: آپلود ویدیو، مدیریت کانال، مشاهده آمار.

- ادمین: مدیریت کاربران، مشاهده گزارش‌ها، تنظیمات شبکه.

- سرویس‌های خارجی: احراز هویت OAuth، پخش ویدیو از CDN.

سطح 1 (Data Flow Diagram Level 1)

هدف: شکستن فرآیند اصلی به چندین فرآیند کلیدی و نشان دادن جریان داده بین آنها و دیتا استورها.

فرآیندهای سطح 1:

1. مدیریت کاربر (User Management): ثبت‌نام، ورود، FA2، مدیریت پروفایل.
2. مدیریت ویدیو (Video Management): آپلود، پردازش، ذخیره‌سازی متادیتا.
3. پخش ویدیو (Video Playback): درخواست استریم، دریافت از CDN.
4. تعاملات (Interaction): لایک، کامنت، سابسکرایب.
5. مدیریت کانال (Channel Management): پروفایل، پلی‌لیست، لیست ویدیوها.
6. مدیریت ادمین (Admin Management): کاربران، گزارش‌ها، شبکه.
7. جستجو و پیشنهاد (Search & Recommendation): نمایه‌سازی، پیشنهاد ویدیو.

دیتا استورها:

- D1: ذخیره کاربران (SQL)
- D2: ذخیره ویدیو متادیتا (NoSQL)
- D3: ذخیره فایل‌های ویدیو و تامب‌نیل (Object Storage)
- D4: کش (Redis)
- D5: لاگ‌ها و گزارش‌ها (ممکن است در NoSQL)