



อินสตาแกรม

**(Instagram)** เป็นผู้  
ให้บริการในการแบ่งปัน  
รูปภาพและคลิปสั้น ๆ และ  
แบ่งปัน ผ่าน **social**  
**network** โดยให้บริการ  
กับผู้ให้บริการ ทั่วโลก

# INSTAGRAM EVERYDAY

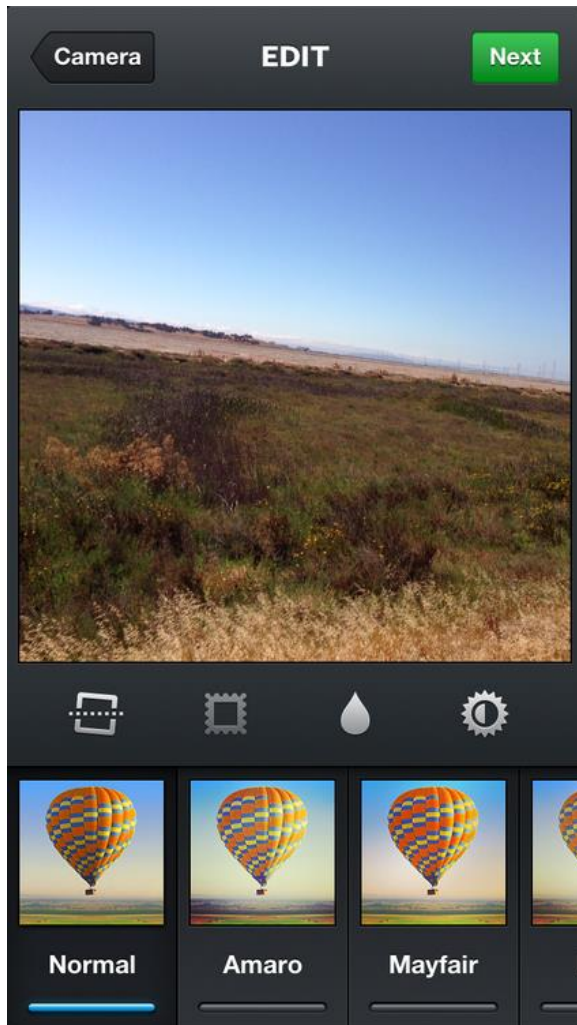
400 Million Users

4+ Billion likes

100 Million photo/video uploads

Top account: 110 Million followers

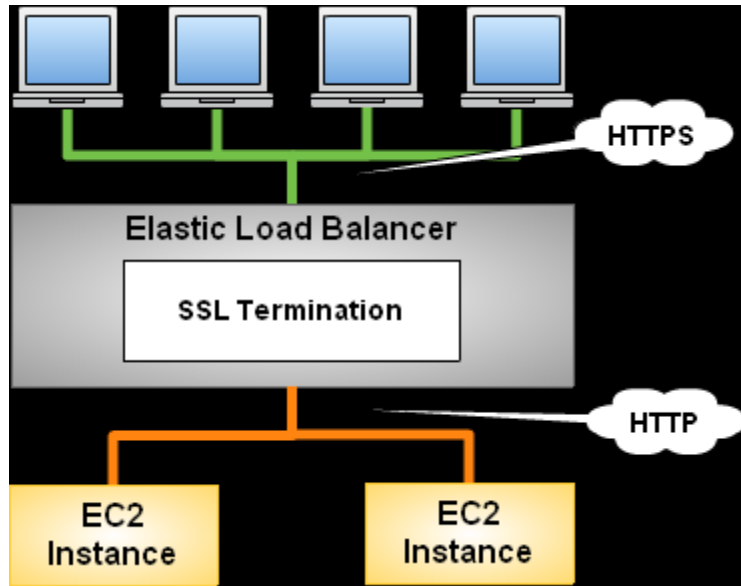
- จำนวนผู้ใช้ในแต่ละวันของ **instagram** แต่ช่วงเวลาที่เป็นเวลาที่คนเล่นเยอะๆ เป็นพิเศษเช่นวันเสาร์-อาทิตย์หรือวันหยุด ทำให้ มีการ **request** ไปที่ **server** จำนวนมากทำให้ **cpu load** นั้นสูง



จำนวนคนโหลดรูปภาพเยอะเกินไปในแต่ละวัน จึงต้องมีฐานข้อมูลที่พร้อมใช้งานตลอดเวลา และเมื่อมีการใช้งานเยอะๆ จึงทำให้ฐานข้อมูลสามารถล่มได้

ในการบริการของ **instagram** ทำได้หลายอย่างเช่น อัปโหลด รูปคนอื่นภาพ เพื่อให้ไม่เกิดการรอและสามารถทำได้หลายอย่างในเวลาเดียวกัน

# เทคนิควิธี หรือสถาปัตยกรรมที่ใช้ในการแก้ปัญหา



- เดิมที Instagram ใช้ server เป็น nginx 2 ตัว สลับกันแบบ DNS round balancer ภายหลัง เปลี่ยนมาใช้ Amazon's Elastic Load Balancer, with 3 NGINX instances ที่สามารถ สลับ กระจาย traffic เข้า-ออกได้ และ ใช้เทคนิคตัดการเชื่อมต่อแบบ SSL ที่ตัว ELB เพื่อลดโหลดของ NGINX ใช้ [Amazon Route53](#) สำหรับ DNS
  - ซึ่งสามารถลด cpu load ของ nginx



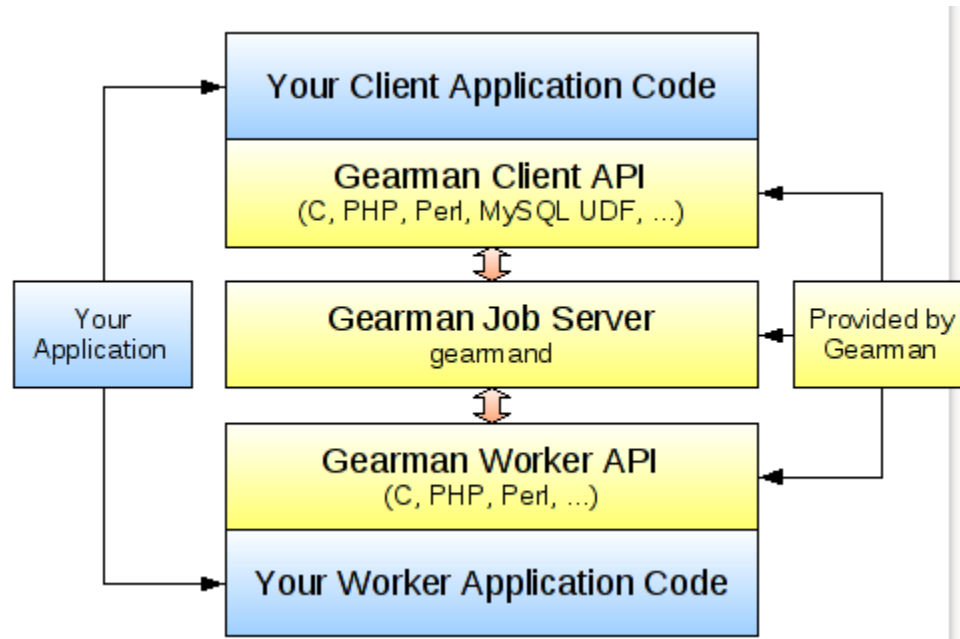
ใช้ Django เป็น application server (ภาษา Python)  
เครื่องที่รันเป็นเครื่องรองรับโหลดหนักพิเศษของ EC2 คือรุ่น High-CPU Extra-Large (ใช้ 25 instance)  
เพราะลักษณะงานของ Instagram เน้นซีพียูมากกว่าแรม จึงเลือกรุ่น High-CPU Extra-Large และช่วยในการรับ request ได้มากขึ้น



- ข้อมูลส่วนใหญ่เก็บใน PostgreSQL
- เครื่องที่รัน PostgreSQL ทุกเครื่องจะต่อแบบ master-replica โดยใช้ Streaming Replication
  - มีผล ทำให้ช่วยในการสำรองข้อมูล เพื่อ database ตัวแรกล่ม ก็มีอีกตัวในการทำงานต่อได้เลย
- รูปภาพทั้งหมดเก็บอยู่บน Amazon S3
- ใช้ Amazon CloudFront เป็น CDN ช่วยกระจายโหลดรูปภาพ
  - ลด load ที่ server หลักในการส่งรูป โดย CDN จะช่วยในการเก็บ รูปภาพที่เป็น static file และส่งถึงผู้ใช้ ที่ใกล้เคียงได้เร็ว



- เครื่องที่ใช้รันฐานข้อมูลเป็นรุ่น **Quadruple Extra-Large memory** จำนวน 12 instance
  - งานเกือบทั้งหมดรันบนหน่วยความจำตลอดเวลา เพื่อประสิทธิภาพที่ดี เพราะระบบการเก็บข้อมูลบนดิสก์ของ **Amazon** ตอบสนองช้าเกินไป



## Gearman สำหรับการจัดคิวงานแบบ asynchronous

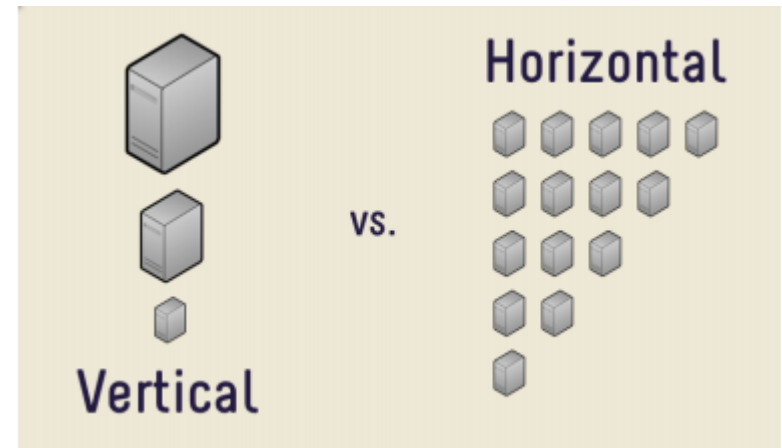
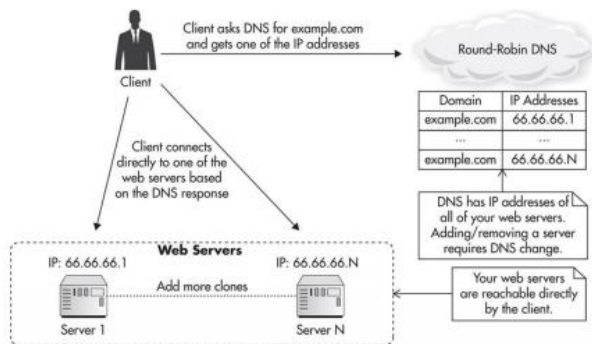
-งานพวกอัปโหลดไปยัง **social network** อื่น เช่น **facebook** หรือ **twitter** จะรันอยู่เบื้องหลังผ่านระบบคิวงานนี้แยกจากการอัปโหลดรูป ทำให้ไม่ต้องรอรูปที่จะอัปไปยัง **social network** อื่น สามารถอัปโหลดรูปใน **instagram** ได้ทันที



# ความสัมพันธ์กับเนื้อหา ที่เรียนใน ชม. บรรยาย

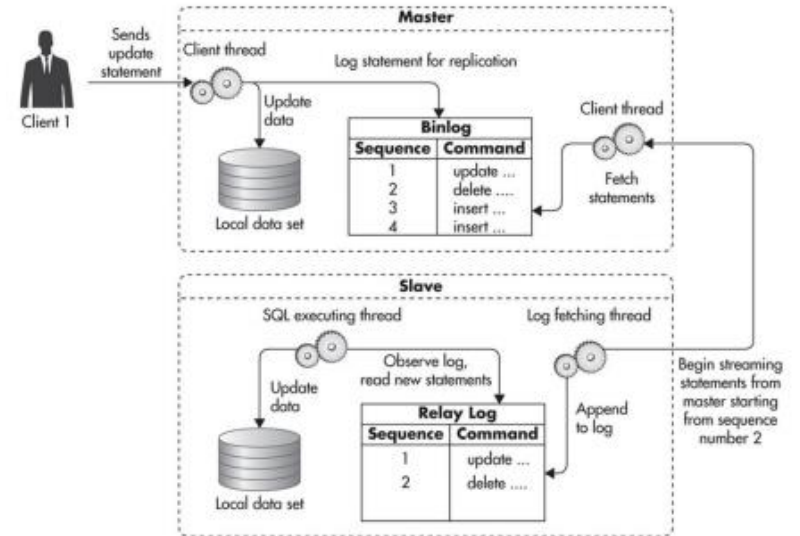
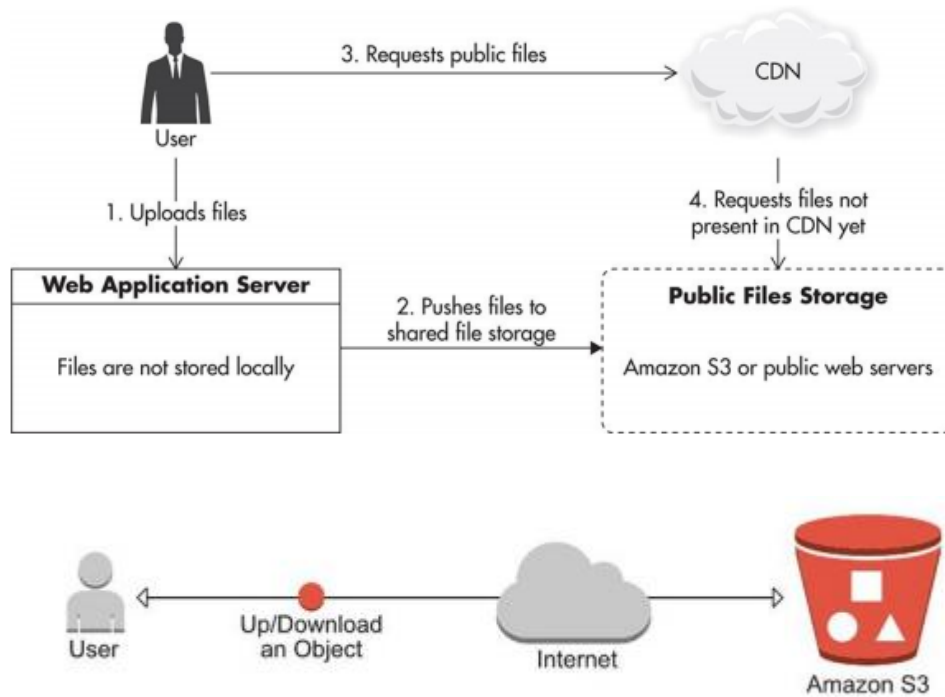
## LOAD BALANCERS

- Once your clients resolve your domain name to an IP address using a DNS service, they will need to connect to that IP to request the page or web service endpoint.



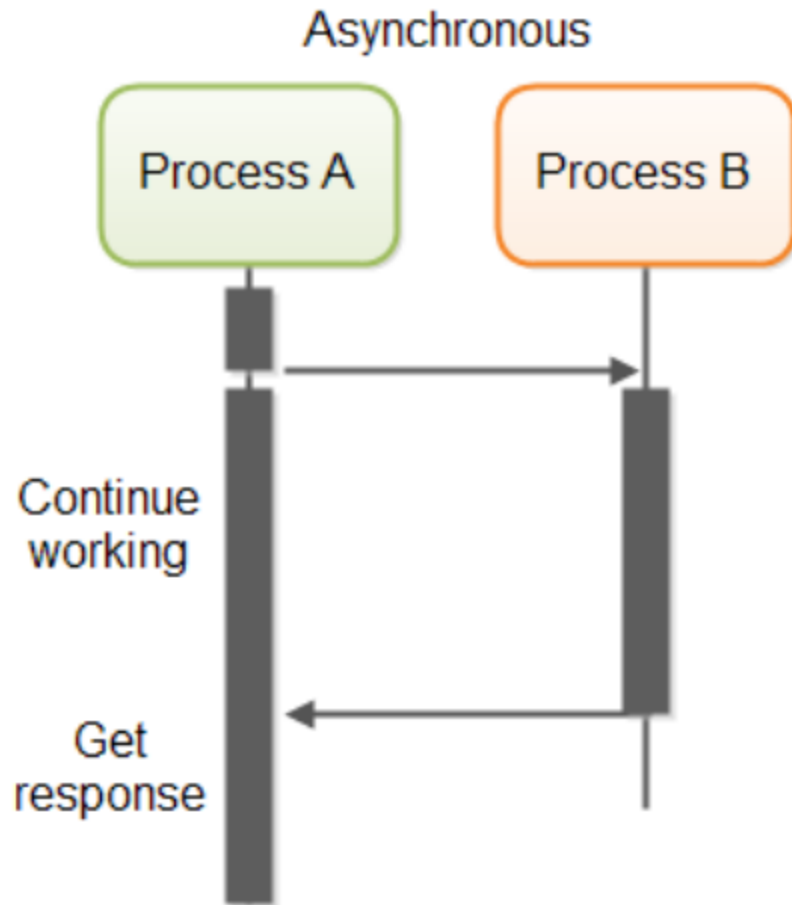
1.การทำ load balancers

2. การเพิ่มประสิทธิภาพแบบ vertical และ horizontal



3. การทำ CDN ในการ cache static file โดย รูปจะเก็บอยู่ใน amazon s3

4. การทำ MASTER-SLAVE REPLICATION ช่วยในการสำรองข้อมูล



5. ASYNCHRONOUS PROCESSING ช่วยในการทำงานที่ซ้อนกันโดยไม่ต้องรอให้อีก **process** ทำงานเสร็จ

## แหล่งอ้างอิง

- <https://www.blognone.com/node/31590>
- <http://highscalability.com/blog/2012/4/9/the-instagram-architecture-facebook-bought-for-a-cool-billio.html>
- <https://www.infoq.com/presentations/instagram-scale-infrastructure>
- <https://engineering.instagram.com/>