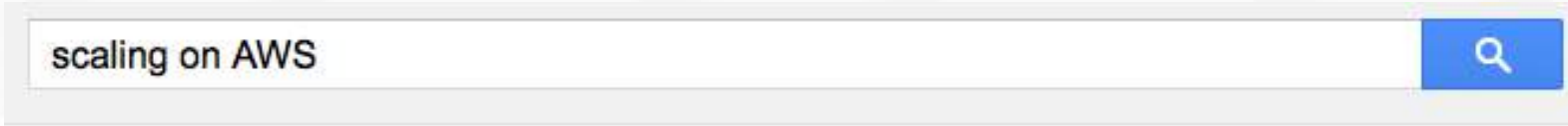
9° Foro de Cloud Computing:
Arquitectura Cloud

FOROS ISIS

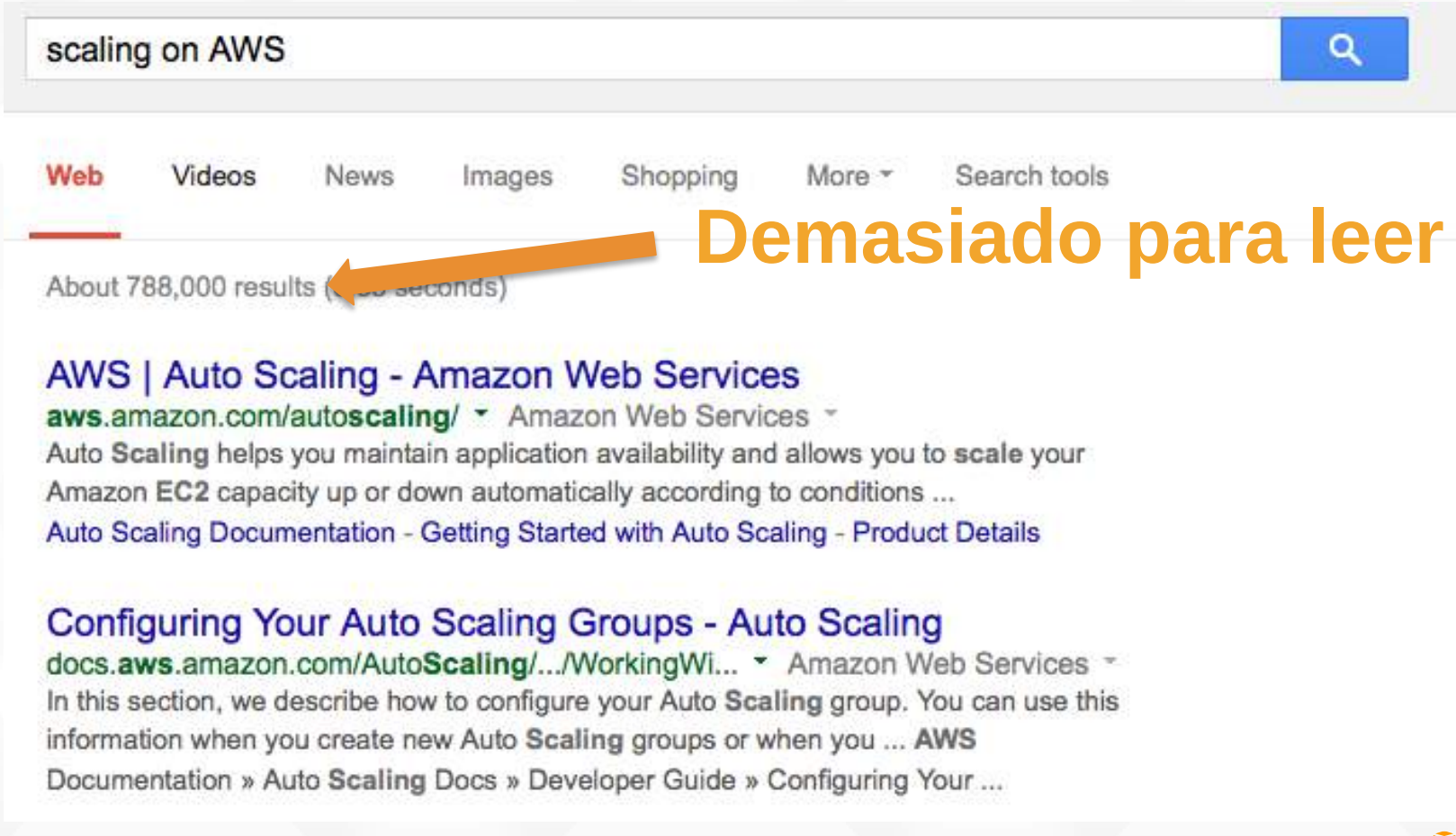
Escalando para sus primeros 10 Millones de usuarios

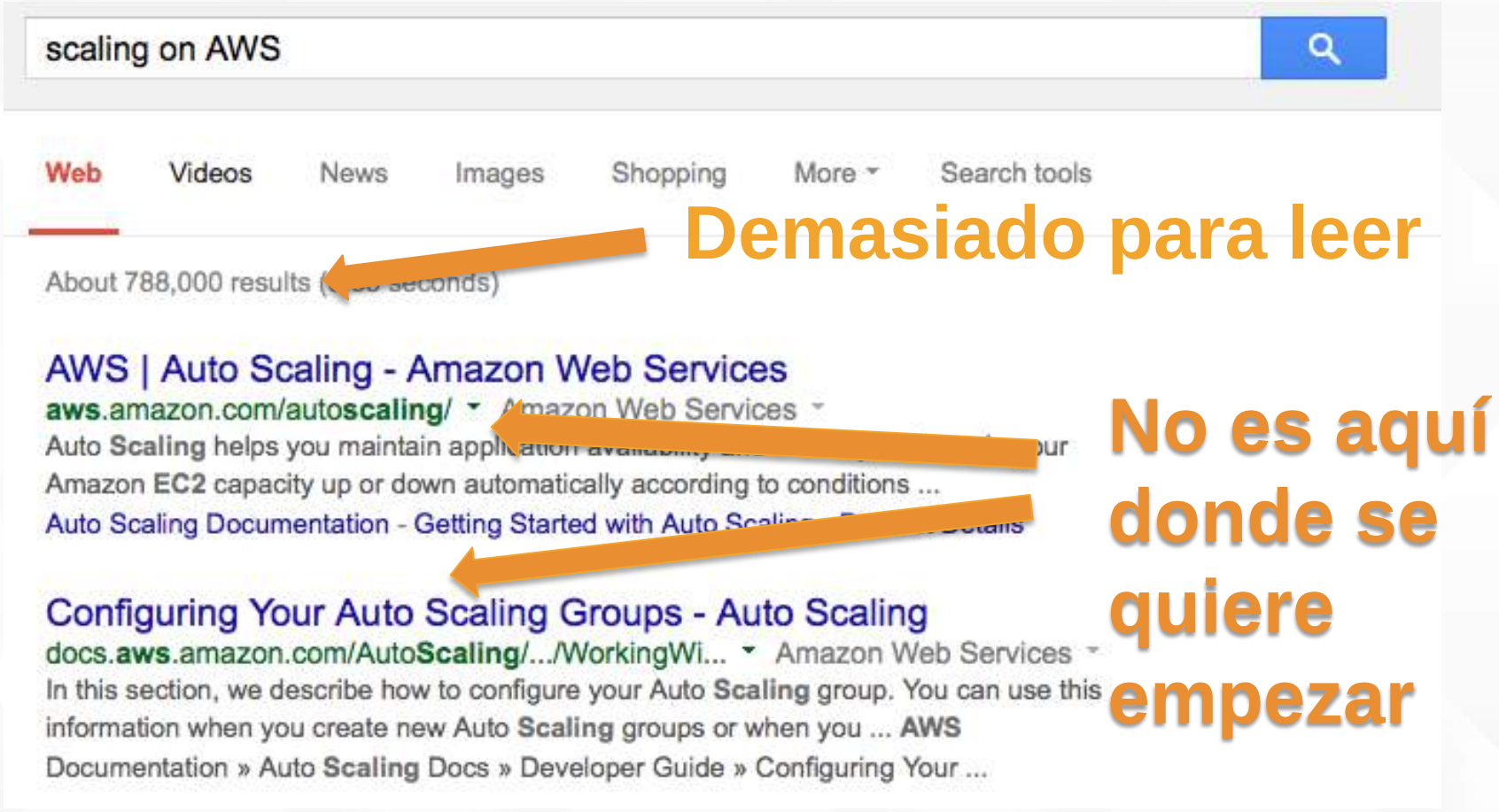
Henry Alvarado
AWS Solutions Architect

Entonces, cómo escalo?



amazon web services


scaling on AWS
Web
Videos
News
Images
Shopping
More
Search tools
Demasiado para leer
About 788,000 results
AWS | Auto Scaling - Amazon Web Services
aws.amazon.com/autoscaling/ Amazon Web Services
Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions ...
Auto Scaling Documentation - Getting Started with Auto Scaling - Product Details
Configuring Your Auto Scaling Groups - Auto Scaling
docs.aws.amazon.com/AutoScaling/.../WorkingWi... Amazon Web Services
In this section, we describe how to configure your Auto Scaling group. You can use this information when you create new Auto Scaling groups or when you ... AWS Documentation » Auto Scaling Docs » Developer Guide » Configuring Your ...



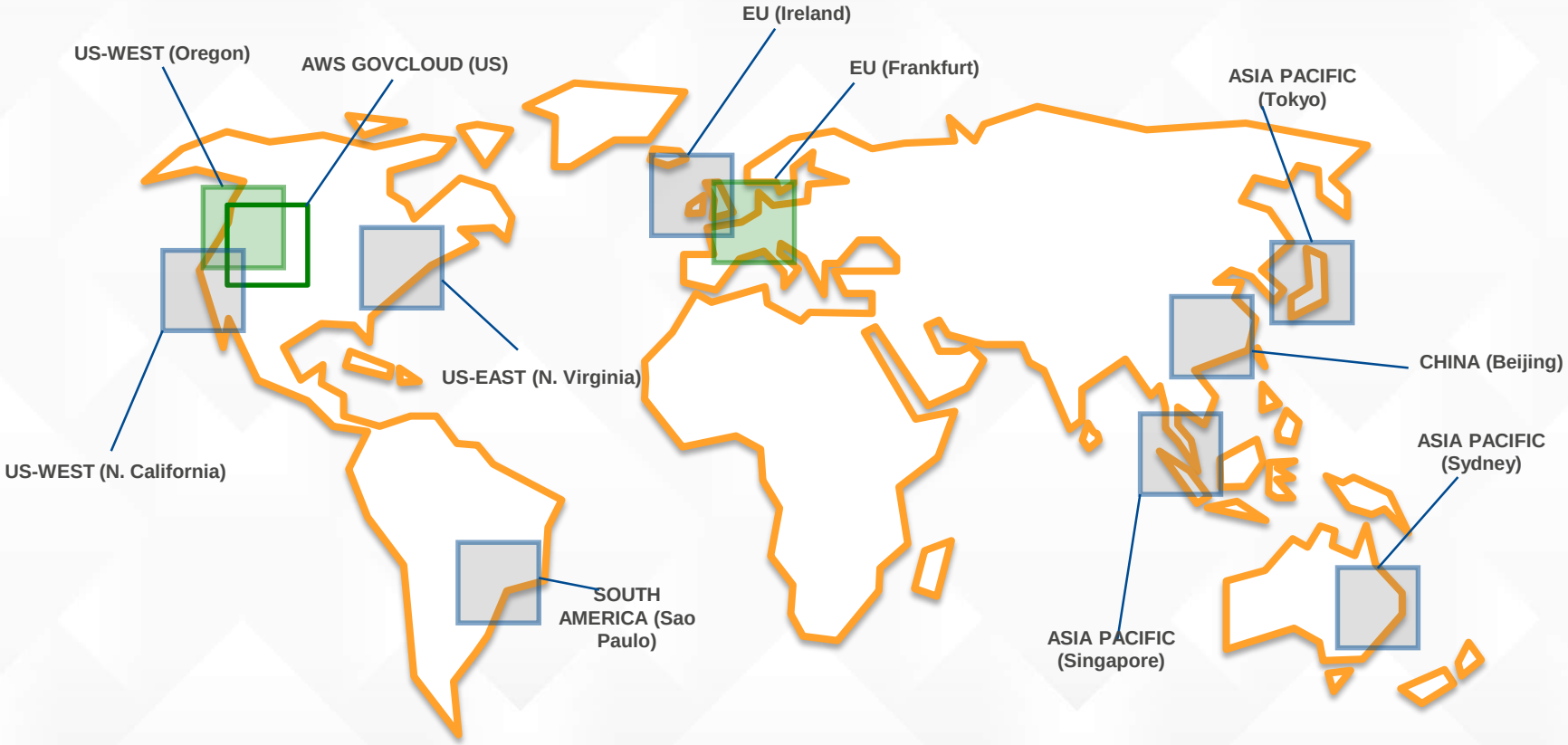
amazon web services

Auto Scaling es una herramienta y un objetivo. No es una única cosa que arregla todo.

Qué necesitamos primero?

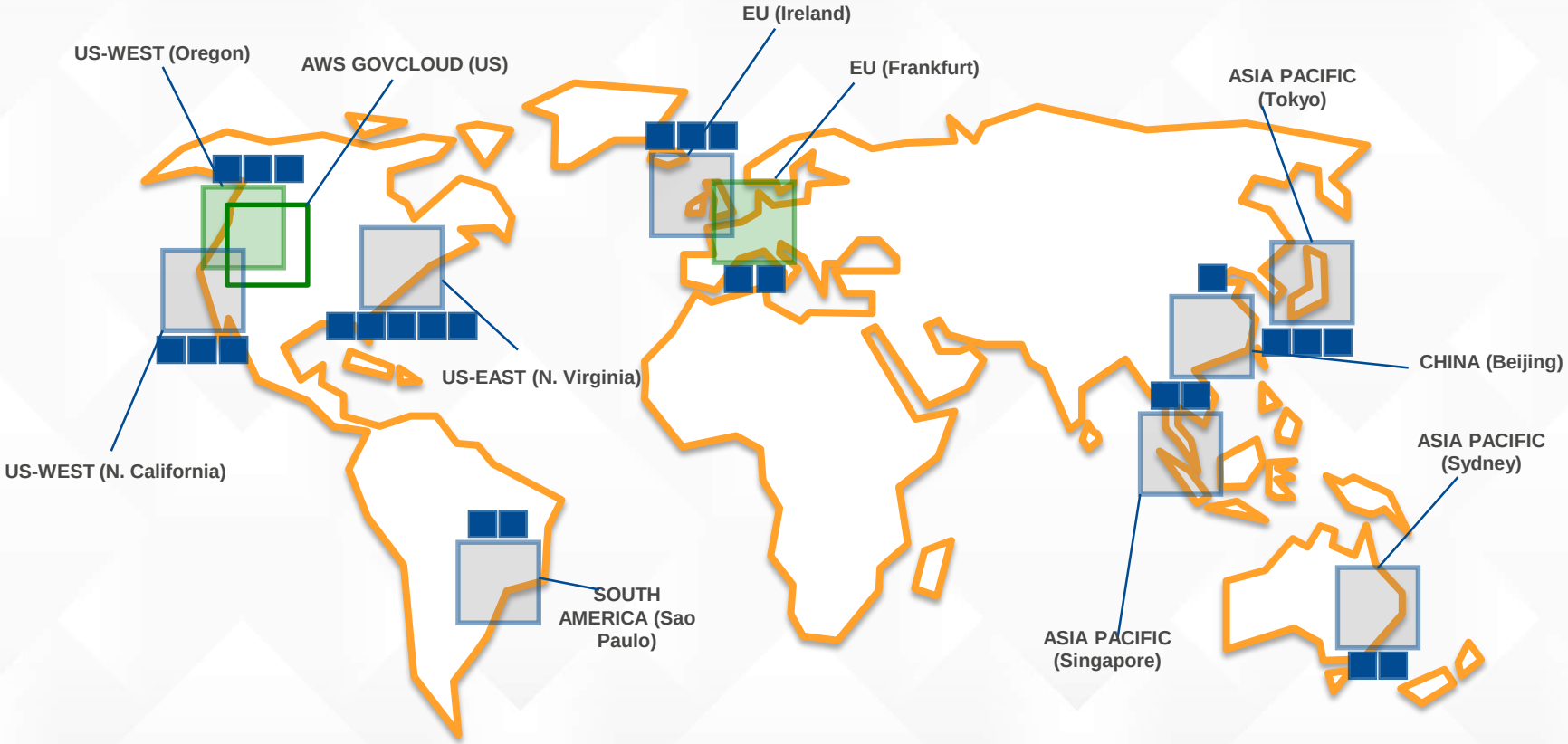
Algunos conceptos básicos…

Regiones



amazon web services

Zonas de disponibilidad



amazon web services

Edge locations

amazon web services

Applications

Virtual Desktops

Collaboration and Sharing

Platform services

Databases

- Relational
- No SQL
- Caching

Analytics

- Hadoop
- Real-time
- Data Warehouse
- Data Workflows

App Services

- Queuing
- Orchestration
- App Streaming
- Transcoding
- Email
- Search

Deployment & Management

- Containers
- Managed User Directories
- Dev/ops Tools
- Resource Templates
- Usage Tracking
- Monitoring and Logs

Mobile Services

- Identity
- Sync
- Mobile Analytics
- Notifications

Foundation services

Compute
(VMs, Auto Scaling and Load Balancing)

Storage
(Object, Block and Archive)

Security and Access Control

Networking

Infrastructure

Regions

Availability Zones

CDN and Points of Presence

amazon web services

AWS building blocks

Servicios altamente disponibles y tolerante a fallas desde su concepción

- Amazon CloudFront
- Amazon Route53
- Amazon S3
- Amazon DynamoDB
- Elastic Load Balancing
- Amazon SQS
- Amazon SNS
- Amazon SES
- Amazon SWF
- …

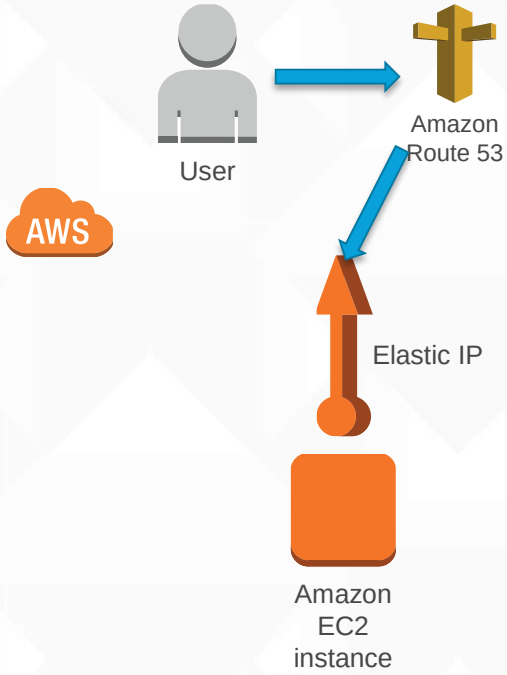
Servicios altamente disponibles con una arquitectura correcta

- Amazon EC2
- Amazon Elastic Block Store
- Amazon RDS
- Amazon VPC

Entonces, empecemos desde el día 1 con el primer usuario (usted)

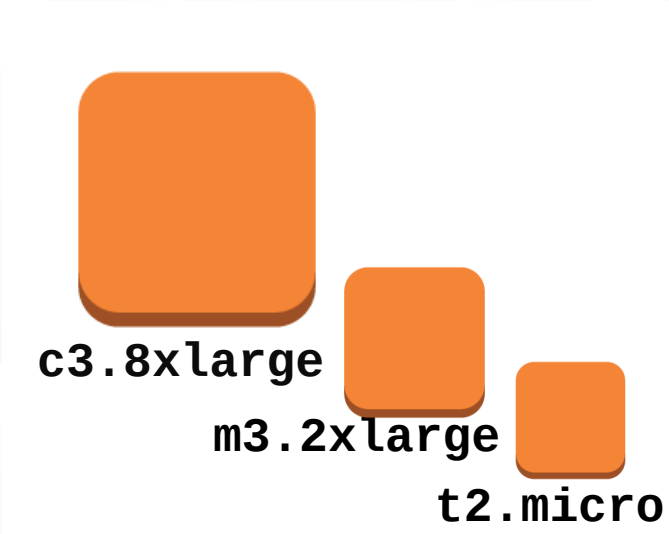
Día 1, usuario 1

- Una única instancia Amazon EC2
  - El stack completo en este host
    - Aplicación web
    - Base de datos
    - Administración
    - Entre otros…
- Una única IP pública
- Amazon Route 53 para DNS



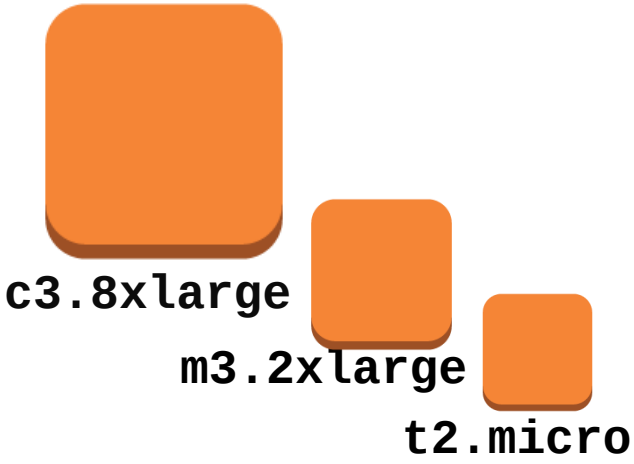
“Vamos a necesitar una caja mas grande”

- Solución más simple
- Posibilidad de usar PIOPS
- Instancias para alto I/O
- Instancias con alta memoria
- Instancias con alto CPU
- Instancias con alto almacenamiento
- Fácil cambio de tamaño de instancia
- Eventualmente llegará al límite



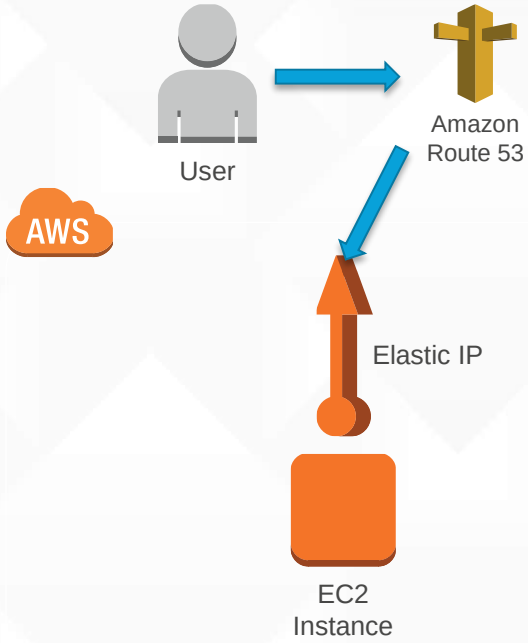
“Vamos a necesitar una caja mas grande”

- Solución más simple
- Posibilidad de usar PIOPS
- Instancias para alto I/O
- Instancias con alta memoria
- Instancias con alto CPU
- Instancias con alto almacenamiento
- Fácil cambio de tamaño de instancia
- **Eventualmente llegará al límite**



Día 1, usuario 1

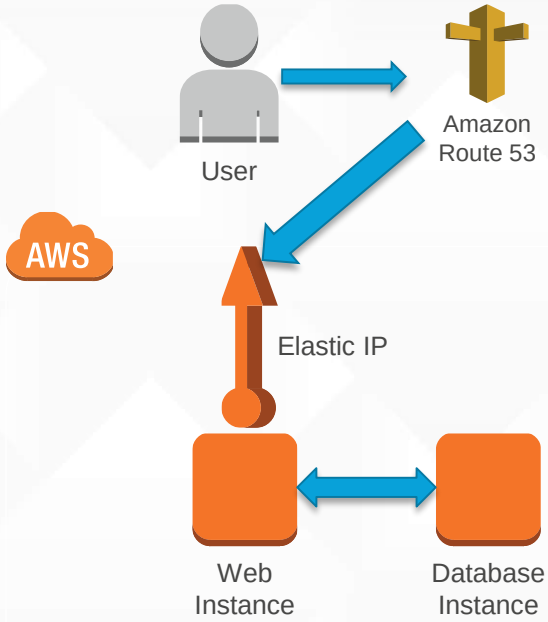
- Potencialmente podríamos atender de algunos cientos a algunos miles dependiendo de la complejidad de la aplicación
- **No hay failover**
- **No hay redundancia**
- **Muchos huevos en la misma canasta**



Día 2, usuario > 1

Primero, separemos nuestro host único en más de uno

- Web
- Base de datos
  - Usar un servicio de base de datos?



Opciones de base de datos

No administrada

Base de datos en Amazon EC2

Su decisión de rodar la base de datos en Amazon EC2

Traiga su propia licencia (BYOL)

Administrada

Amazon RDS

Microsoft SQL Server, Oracle, MySQL, o PostgreSQL como servicio administrado

Licenciamiento flexible: BYOL o licencia incluida

Amazon DynamoDB

Servicio de base de datos NoSQL usando almacenamiento SSD

Escalabilidad simple con cero administración

Amazon Redshift

Servicio de DW de gran escala, masivamente paralelo.

Rápido, poderoso y fácil de escalar

amazon web services

Pero, como escojo la tecnología de DB que necesito? SQL? NoSQL?

A algunos no les va a gustar esto, pero…

Inicie con bases de datos SQL

Por qué comenzar con SQL?

- Es una tecnología establecida y bien conocida
- Existen cientos de libros, comunidades, herramientas, código y más.
- Usted no va a derrumbar una BD SQL con sus primeros 10 millones de usuarios. No, en realidad no lo hará*
- Patrones claros de escalabilidad.

*A menos que usted esté haciendo algo MUY fuera de lo común o tenga cantidades MASIVAS de datos, pero inclusive así, SQL tendrá un espacio en su stack.

Ajá! Usted dijo “cantidades masivas” y yo voy a tener cantidades masivas!

Si su uso es tal que va a generar TB (> 5) de datos en su primer año, O va a tener un flujo de trabajo de intensidad de datos increíbles, usted podría entonces necesitar NoSQL

Por qué otras razones necesitaría NoSQL?

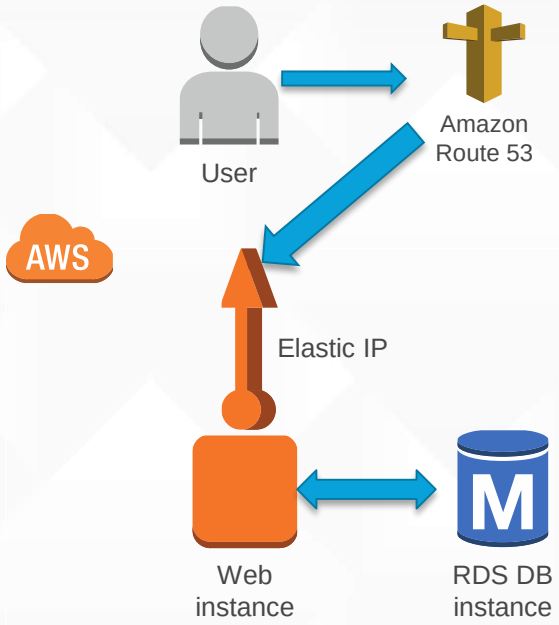
- Aplicaciones de muy baja latencia
- Datasets basados en metadata
- Data altamente no relacional
- Necesidad de constructs de datos sin esquema*
- Cantidades masivas de datos (de nuevo, en el orden de TB)
- Rápida ingestión de datos (miles de records/seg)

*Necesidad != “Es más fácil desarrollar sin esquemas”

Usuario > 100

Primero, separemos nuestro único host en más de uno:

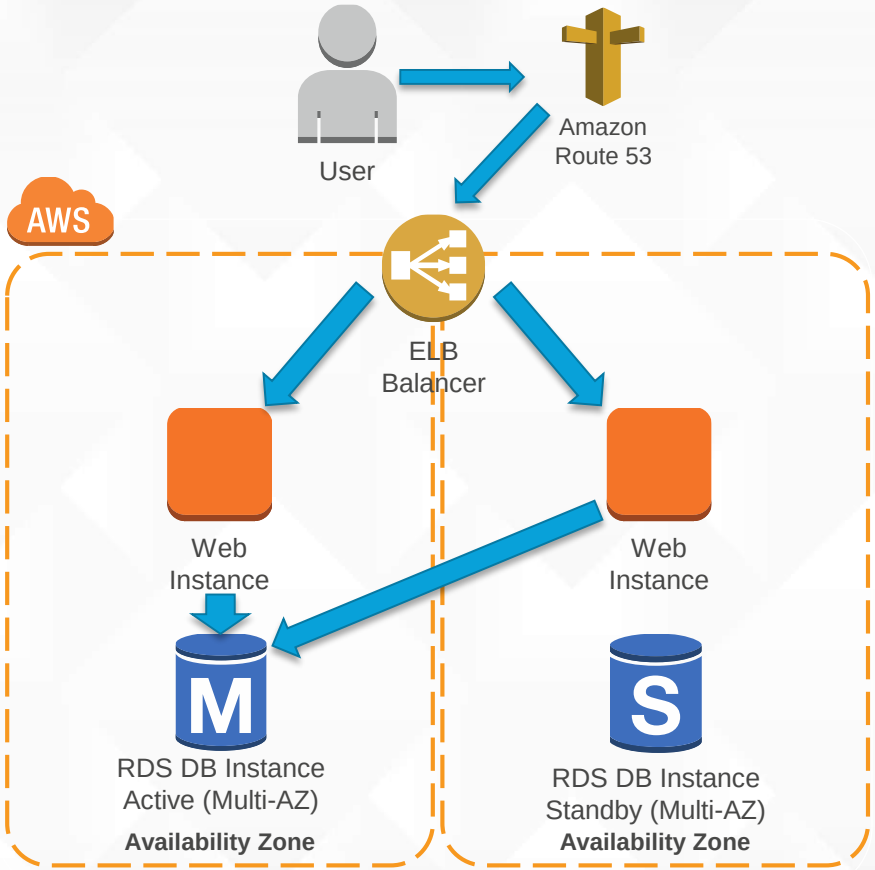
- Web
- Base de datos
  - Use Amazon RDS para hacer su vida más fácil



Usuario > 1000

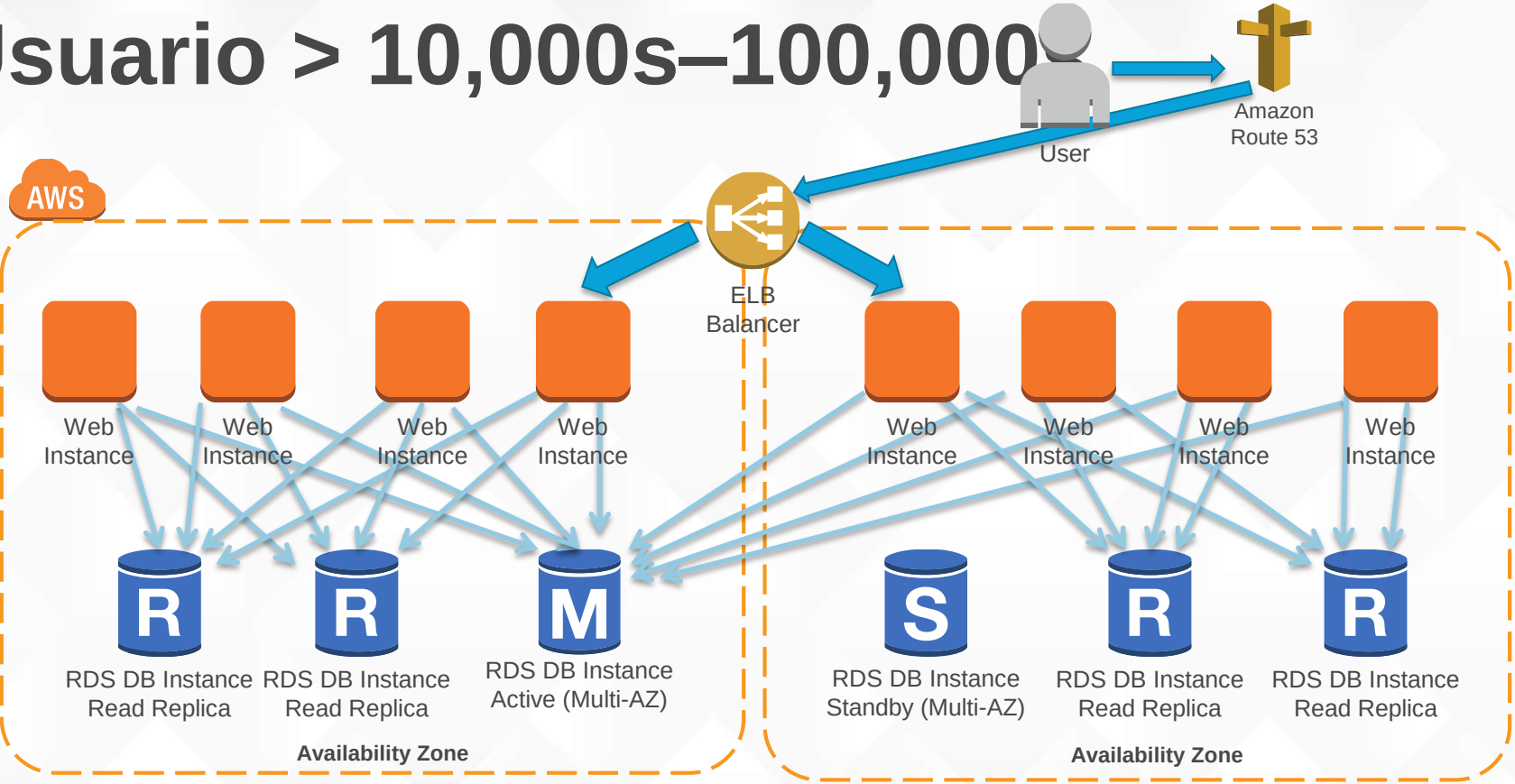
Después, vamos a atacar nuestra falta de failover y problemas de redundancia:

- Elastic Load Balancing (ELB)
- Otra instancia web
  - En otra zona de disponibilidad
- RDS Multi-AZ



Escalando esto horizontalmente y verticalmente nos va a llevar bastante lejos: (decenas a cientos de miles)

Usuario > 10,000s–100,000



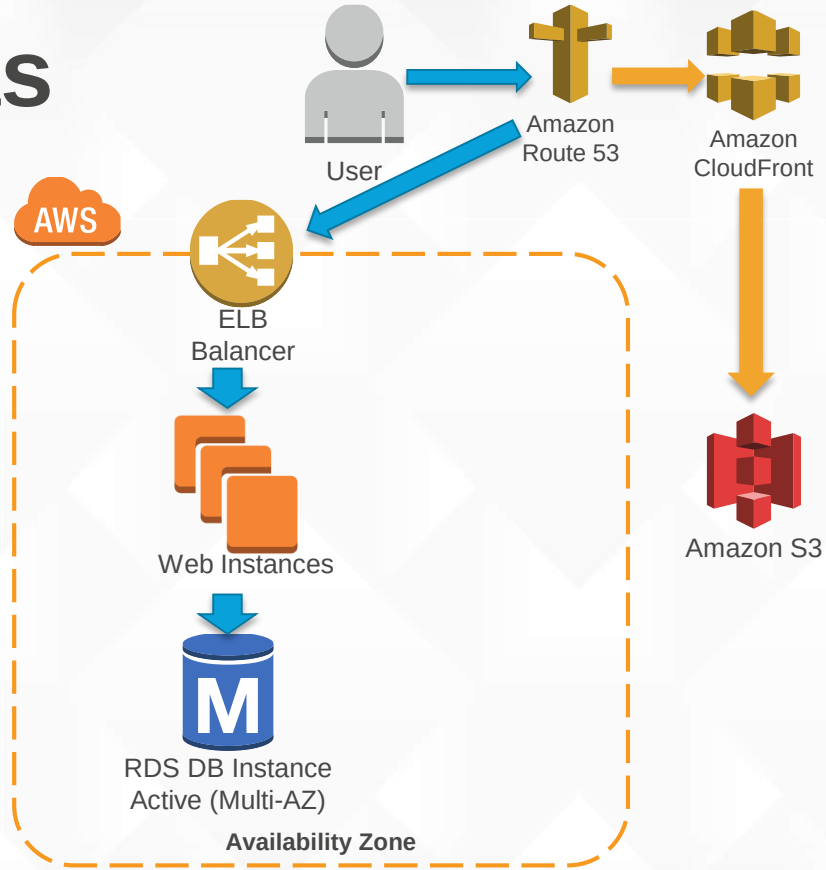
amazon web services

Esto nos llevará lejos, pero nos importa también el *performance* y la *eficiencia,* entonces vamos a mejorarlo un poco más:

Movamos las cargas

Vamos a aligerar la carga en nuestras instancias web y base de datos:

- **Mover el contenido estático de la instancia web a Amazon S3 y Amazon CloudFront**
- Mover la sesiones/estado y crear un caché para la base de datos usando Amazon ElastiCache o Amazon DynamoDB

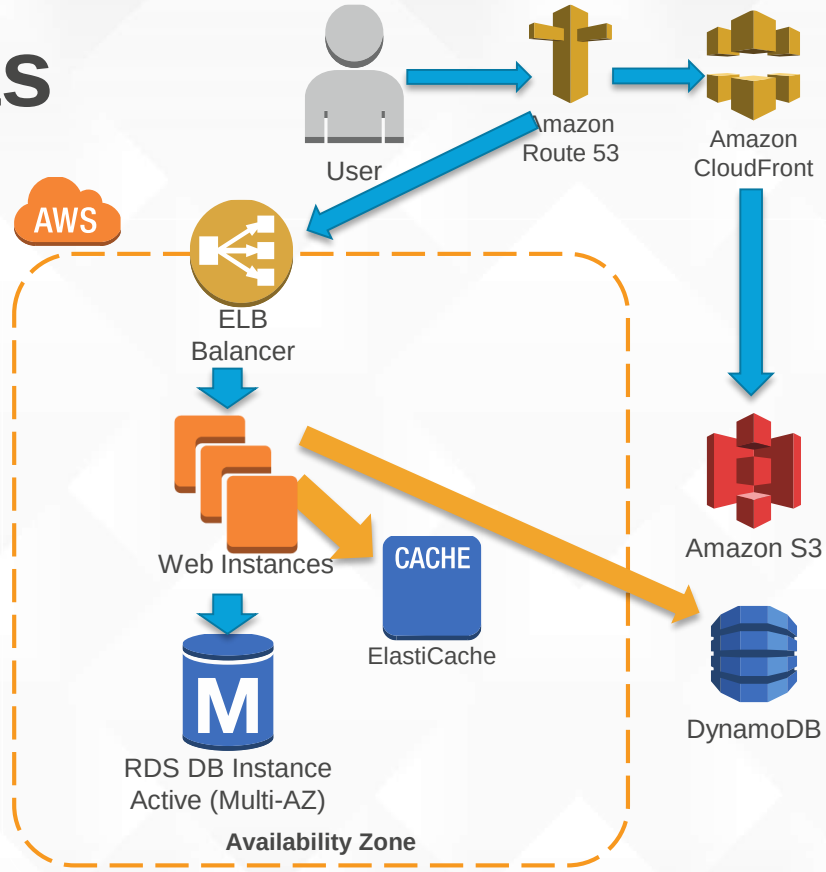


amazon web services

Movamos las cargas

Vamos a aligerar la carga en nuestras instancias web y base de datos:

- Mover el contenido estático de la instancia web a Amazon S3 y Amazon CloudFront
- **Mover la sesiones/estado y crear un caché para la base de datos usando Amazon ElastiCache o Amazon DynamoDB**

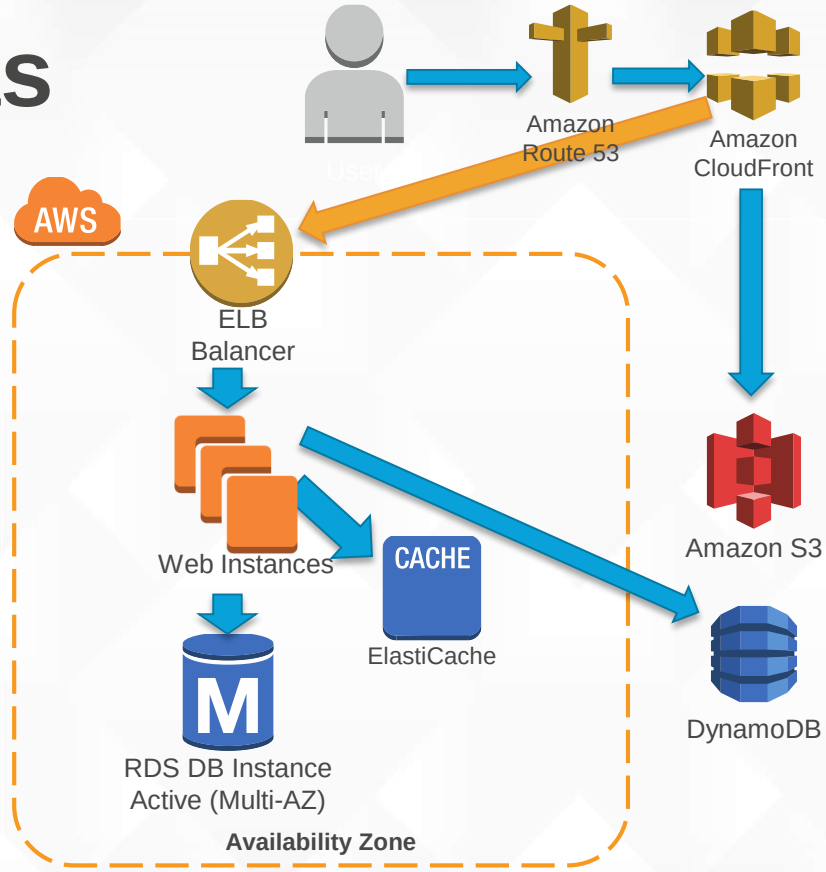


amazon web services

Movamos las cargas

Vamos a aligerar la carga en nuestras instancias web y base de datos:

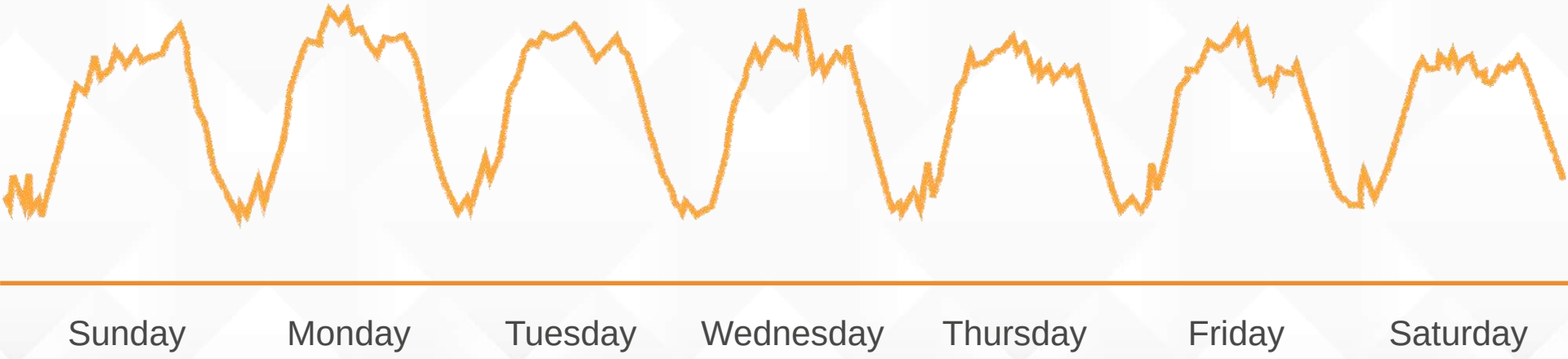
- Mover el contenido estático de la instancia web a Amazon S3 y Amazon CloudFront
- Mover la sesiones/estado y crear un caché para la base de datos usando Amazon ElastiCache o Amazon DynamoDB
- **Mover el contenido dinámico del ELB a Amazon CloudFront**



Ahora que nuestra capa web es mucho más ligera, podemos volver al inicio de nuestra charla…

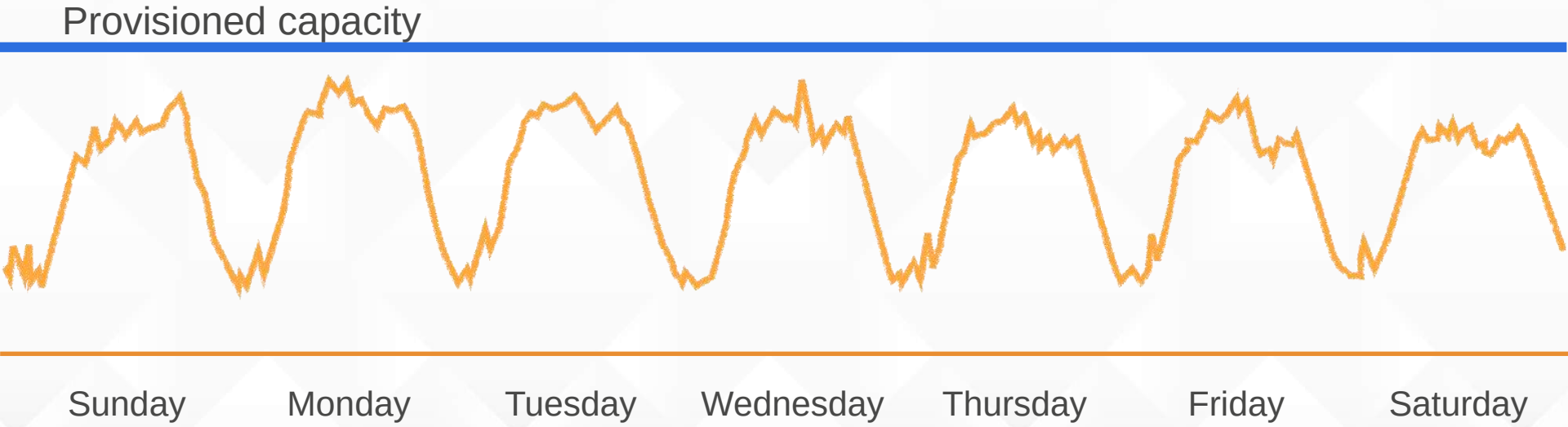
Auto Scaling!

Tráfico típico semanal a Amazon.com



amazon web services

Tráfico típico semanal a Amazon.com



Tráfico en noviembre a Amazon.com

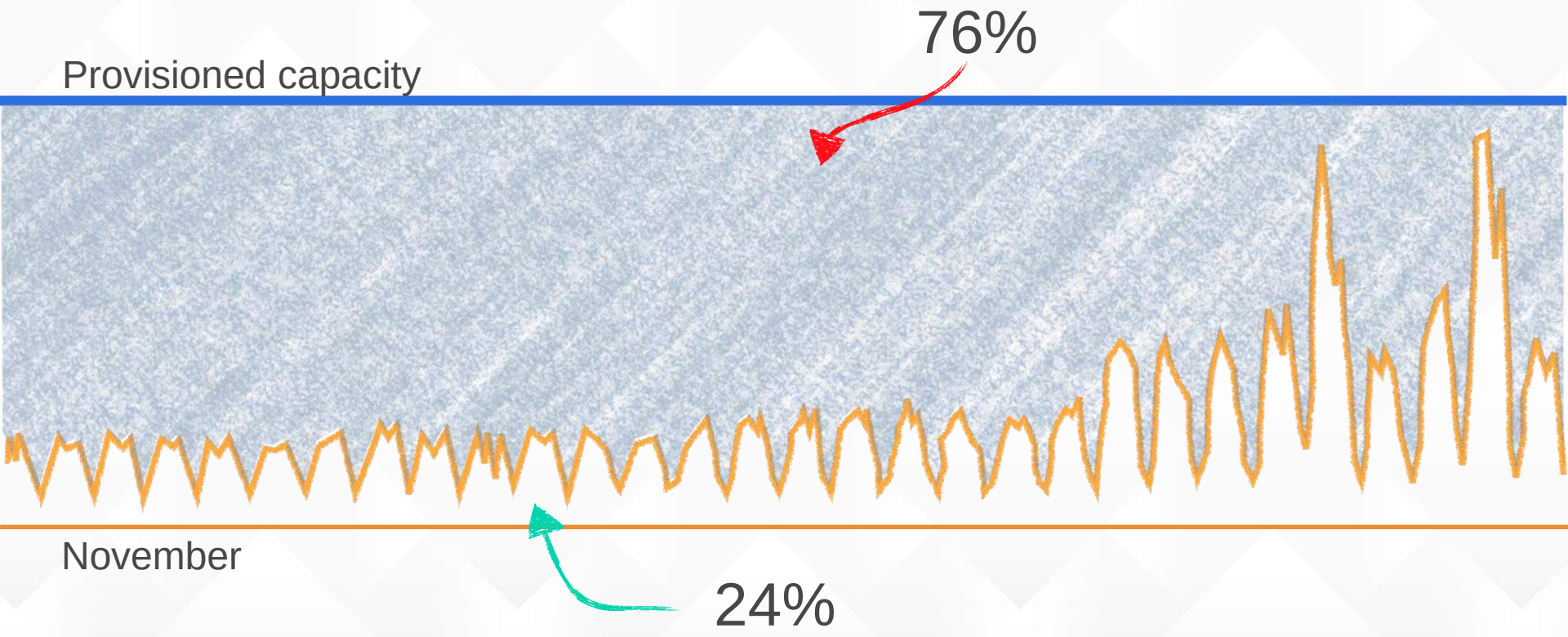
November

amazon web services

Tráfico en noviembre a Amazon.com

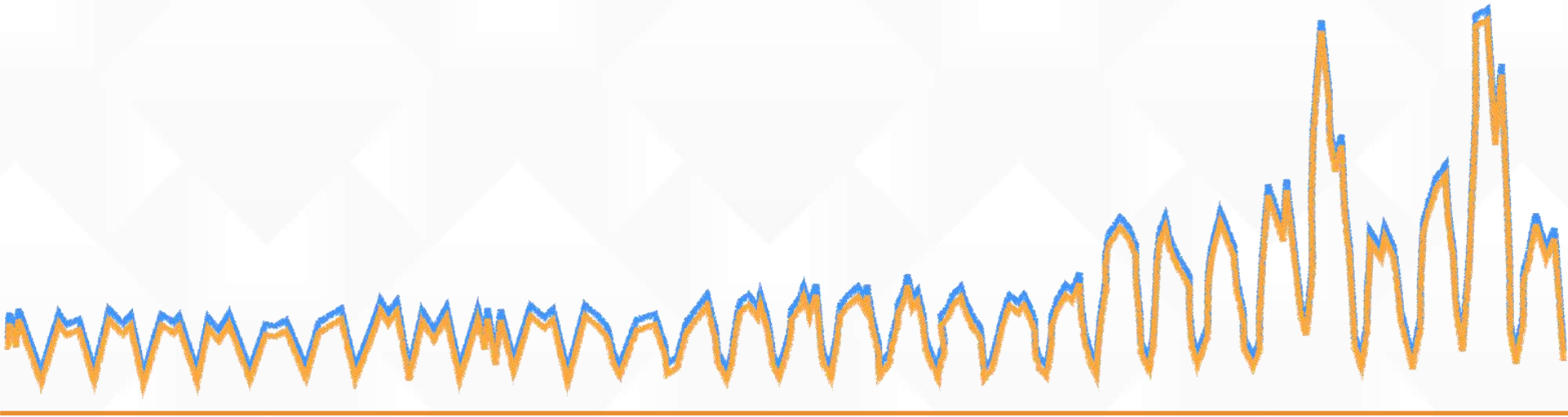
amazon web services

Tráfico en noviembre a Amazon.com



amazon web services

Tráfico en noviembre a Amazon.com



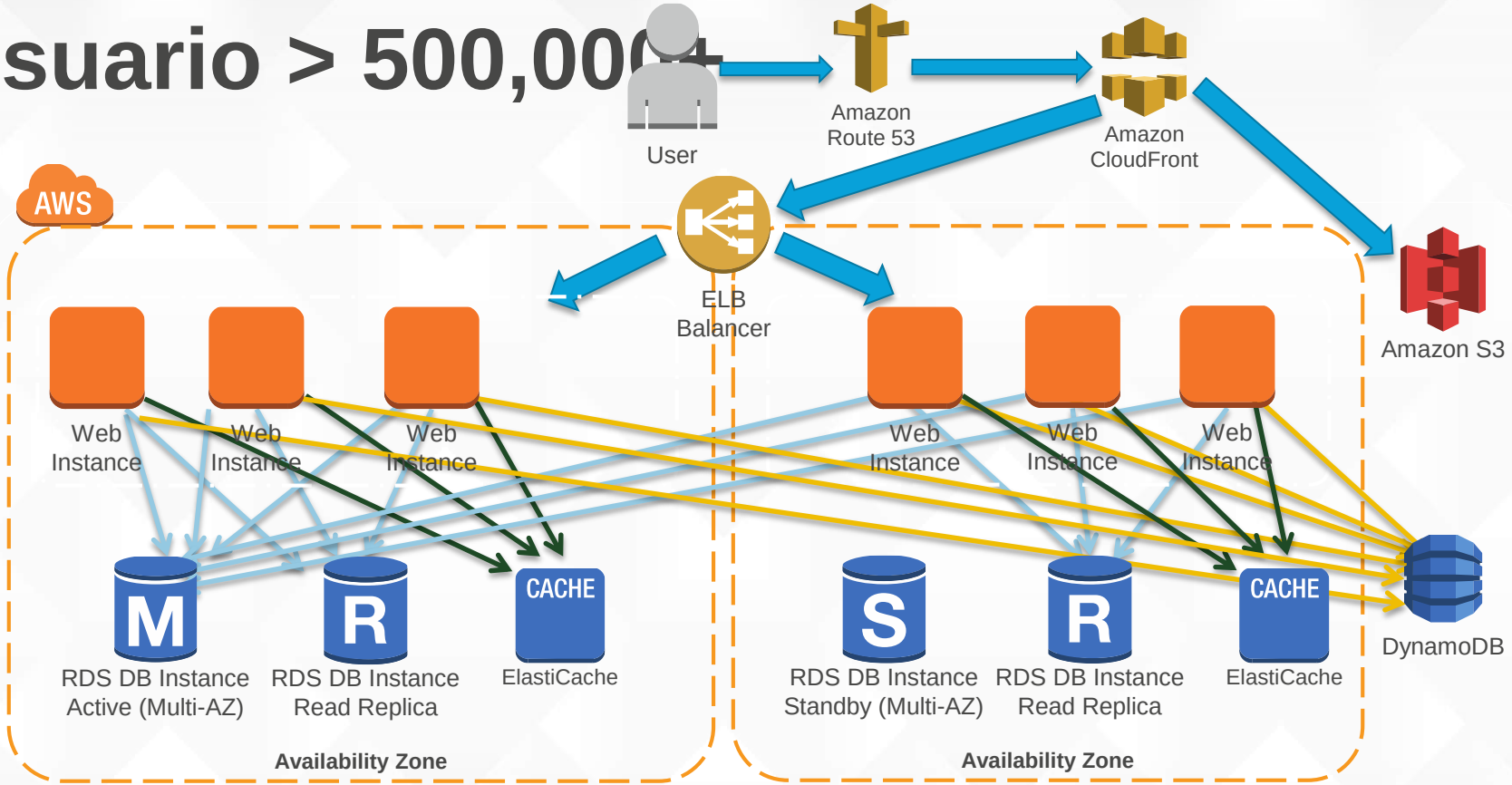
November

amazon web services

Auto Scaling
les permite hacer esto!

amazon web services

Usuario > 500,000+



amazon web services

Use automatización

Administrar su infraestructura va a ser cada día una parte más importante de su tiempo. Use herramientas de automatización para tareas repetitivas:

- Herramientas para administrar sus recursos AWS
- Herramientas para administrar el software y la configuración en sus instancias.
- Automatice el análisis de logs y acciones de los usuarios.

Soluciones de administración de aplicaciones AWS

Servicios de alto nivel

Hágalo usted mismo

AWS
Elastic Beanstalk

AWS
OpsWorks

AWS
CloudFormation

Amazon EC2

Convenience

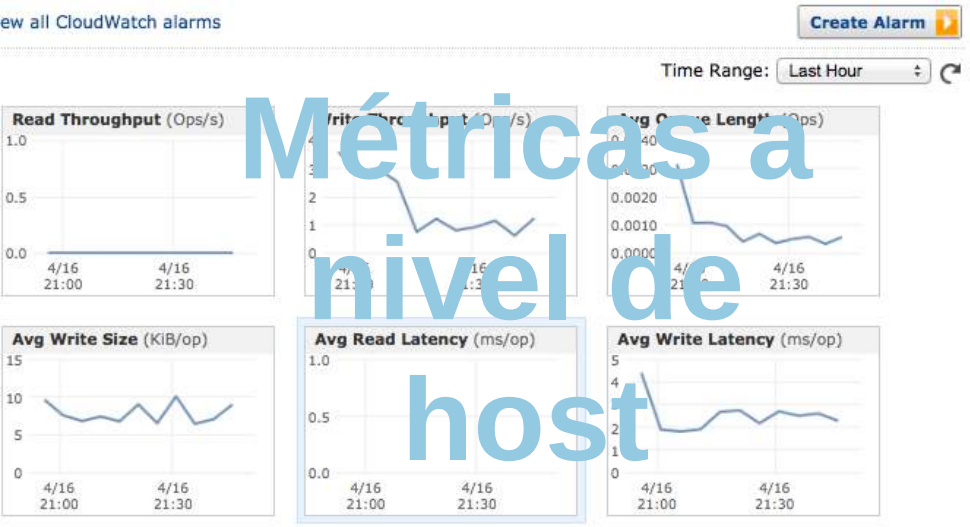
Control

amazon web services

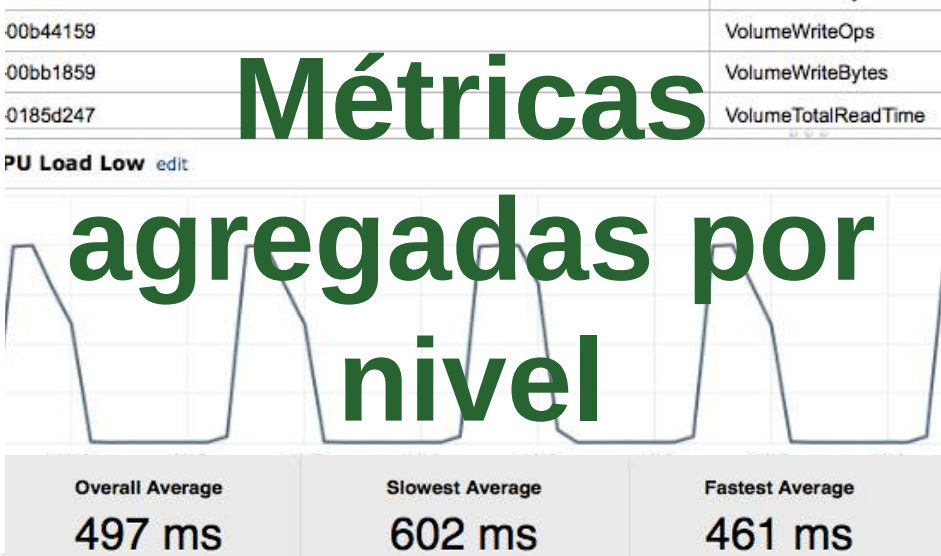
Usuario > 500,000+

Potencialmente usted va a empezar a tener problemas con la velocidad y el performance de sus aplicaciones:

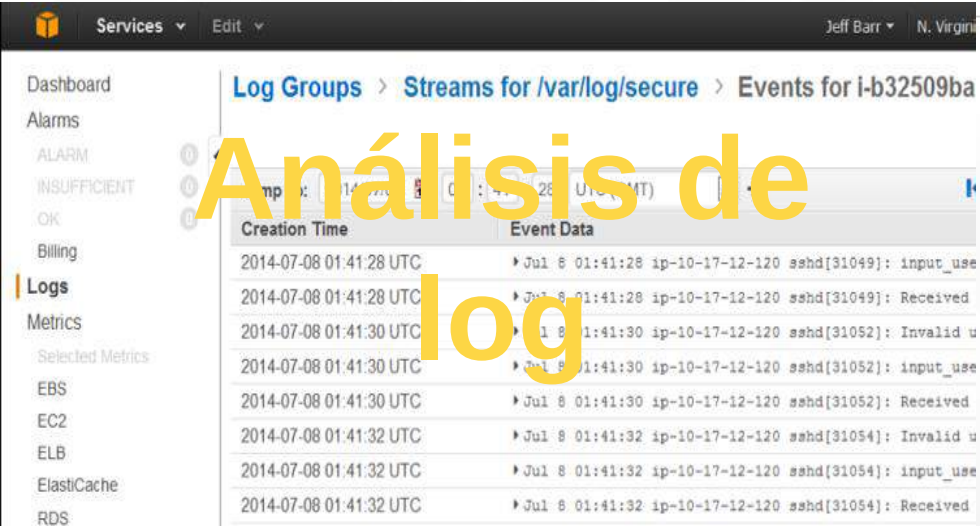
- Asegúrese de tener monitoreo, métricas, alarmas y logs.
  - Si no puede construir una solución interna, use un Third-party como Nagios, NewRelic, entre otros…
- Ponga atención a cuantos clientes hablan bien de su aplicación vs. cuantos no lo hacen y use esta información.
- Intente exprimir la mayor cantidad de performance de cada uno de los servicios o componentes que use.



Métricas a nivel de host



Métricas agregadas por nivel



Análisis de log

Performance externo del sitio

Hay mejoras adicionales a ser realizadas al quebrar su capa de web / aplicación

SOA (Service Oriented Architecture)

- Mover servicios a sus propias capas o módulos. Trate cada uno de ellos como piezas completamente separadas de su infraestructura y escálelas de forma independiente.

- Amazon.com y AWS hacen esto de forma extensiva! Ofrece flexibilidad y un mejor entendimiento de cada uno de los componentes.

Desacoplamiento + SOA = ganador

Si alguien ya creó un servicio que cumple sus necesidades, úselo en vez de construirlo.

No reinvente la rueda.

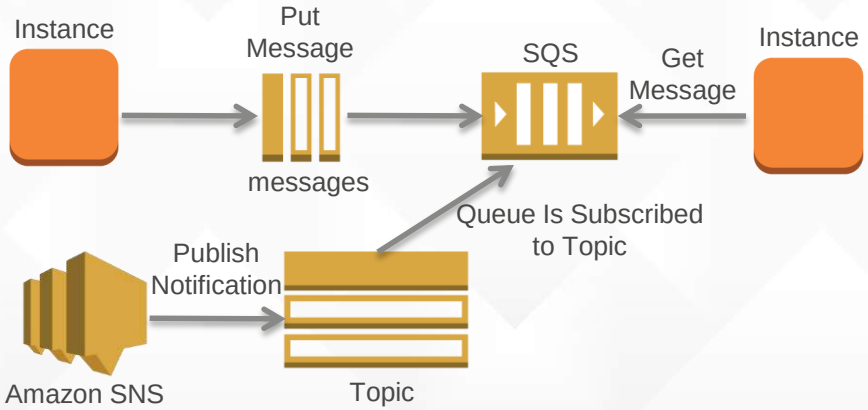
Ejemplos:

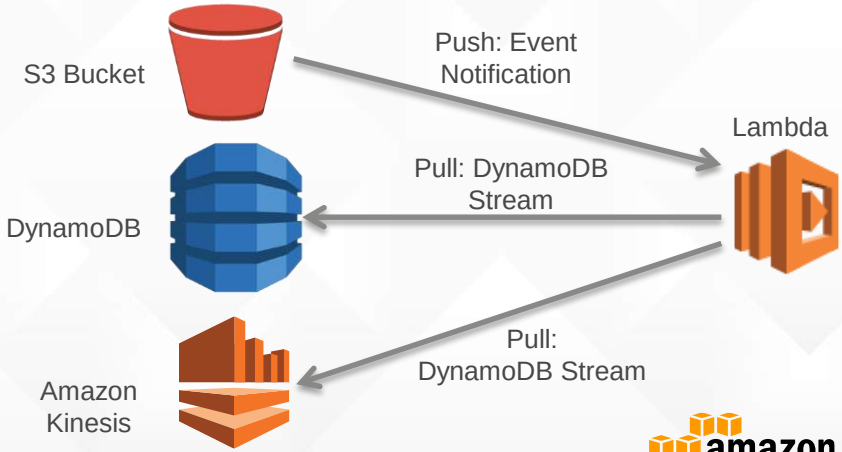
- Email
- Queuing
- Transcoding
- Search
- Databases
- Monitoring
- Metrics
- Logging
- Compute

El desacoplamiento te libera!

Entre más desacoplados, más escalan

- Componentes independientes
- Diseñe todo como una caja negra
- Desacople interacciones
- Favorezca servicios que ya ofrecen redundancia y escalabilidad, en vez de crear sus propios.





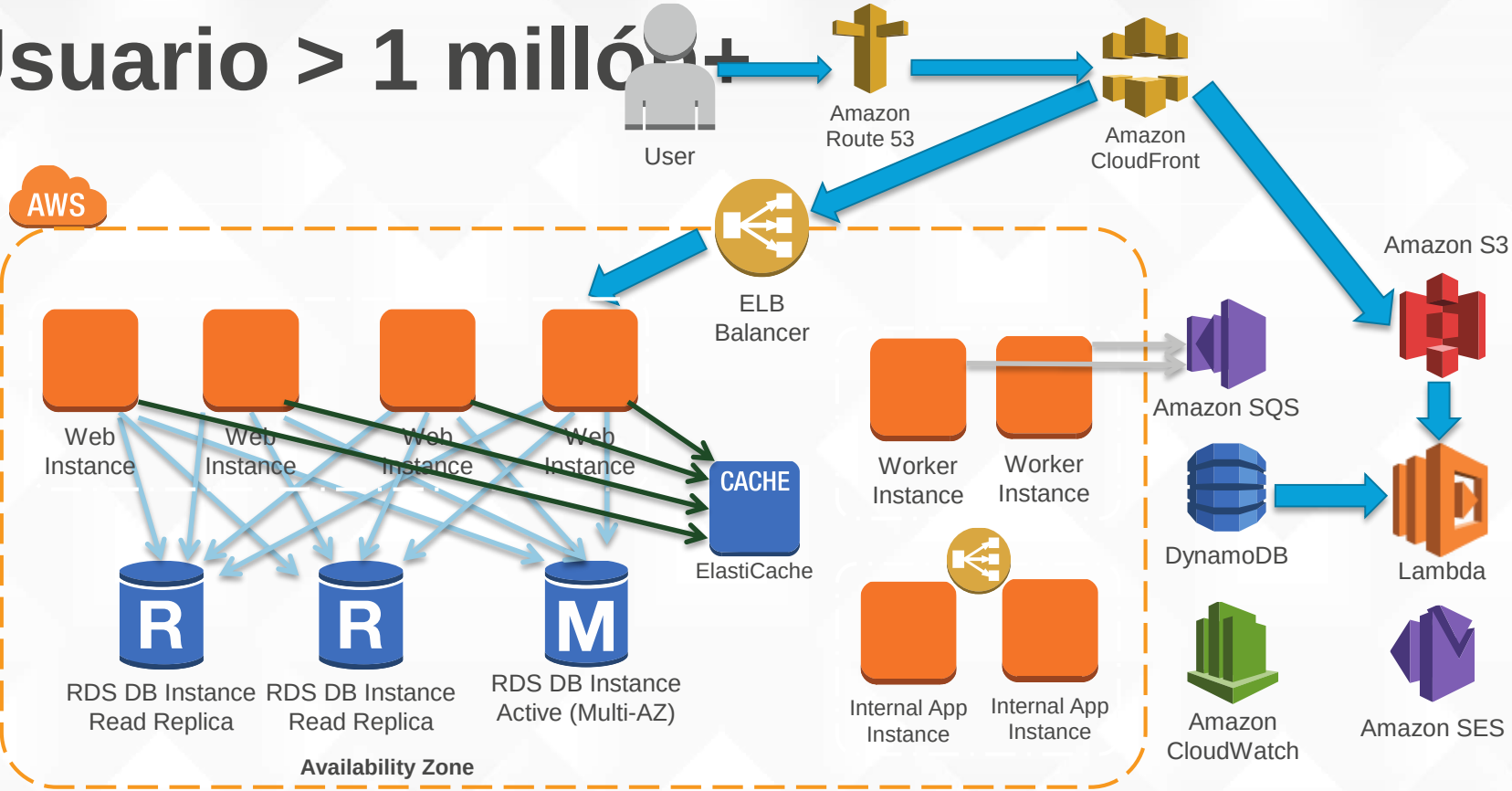
amazon web services

Usuario > 1 millón +

Llegar a millones y más, va a requerir un poco de todo lo que previamente hemos conversado:

- Multi-AZ
- Elastic Load Balancing entre capas
- Auto Scaling
- Service Oriented Architecture
- Sirva contenido de forma inteligente (S3/CloudFront)
- Caché de base de datos
- Remueva el estado de capas que auto escalan

Usuario > 1 millón+



amazon web services

Los siguientes grandes pasos

Usuario > 5 millones – 10 millones

Potencialmente en este punto comenzará a tener problemas con su base de datos sobre conexiones de escrita en la instancia master.

Cómo lo puede resolver?

- Federation — separar en múltiples BDs dependiendo de la función (Foros, Usuarios, Productos..)
- Sharding — separar los datos en múltiples hosts
- Mover algunas funcionalidades a otros tipos de bases de datos (NoSQL, Grafos)

Un breve resumen

Resumen

- Infraestructura Multi-AZ
- Use servicios que escalan por si solos – ELB, Amazon S3, Amazon SNS, Amazon SQS, Amazon SWF, Amazon SES, entre otros.
- Construya con redundancia en todos los niveles.
- Empiece con SQL. En serio!
- Use caché de datos tanto dentro como fuera de su infraestructura.
- Use herramientas de automatización en su infraestructura.

Resumen

- Asegúrese de tener buenas herramientas para métricas/monitoreo/logs.
- Separe capas en servicios individuales (SOA).
- Use Auto Scaling cuando esté listo para ello.
- No intente reinventar la rueda.
- Cámbiese a NoSQL si y cuando tenga sentido.

Poner todo esto junto significa que podrá fácilmente ser capaz de manejar 10+ millones de usuarios!

Hasta el infinito...

Usuario > 10 millones

Iterando sobre estos patrones vistos aquí, podrá llevar su aplicación a más de 100 millones de usuarios

Próximos pasos?

Pase por:

- aws.amazon.com/documentation
- aws.amazon.com/architecture
- aws.amazon.com/start-ups

Comience a usar AWS

- aws.amazon.com/free/

Próximos pasos?

Obtenga ayuda!

- forums.aws.amazon.com
- aws.amazon.com/premiumsupport/
- Su AWS Account Manager
- Un Arquitecto de soluciones

Thank You

amazon web services