Scale, baby, scale!

Julien Simon
Principal Technical Evangelist
Amazon Web Services

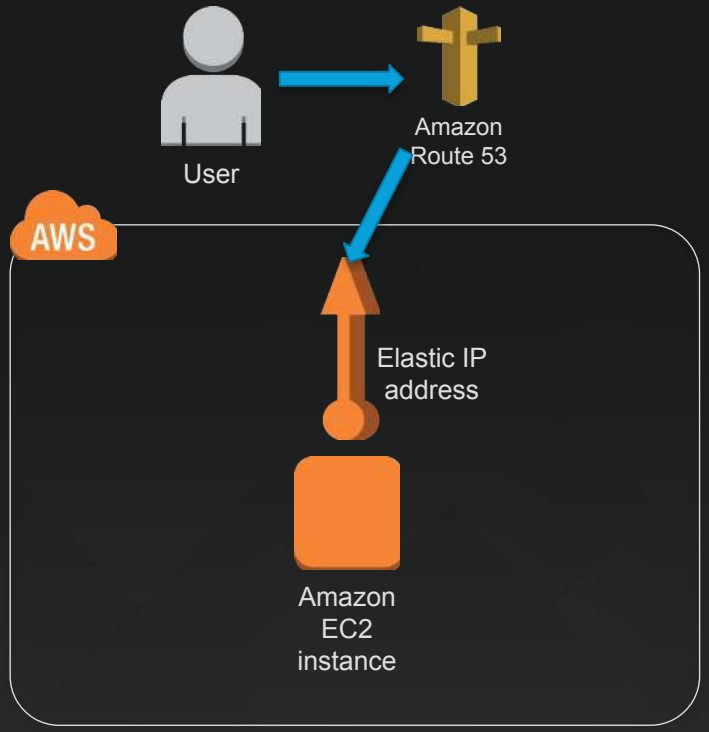
julsimon@amazon.fr
@julsimon

amazon web services

So let’s start from day 1, user 1 (you)

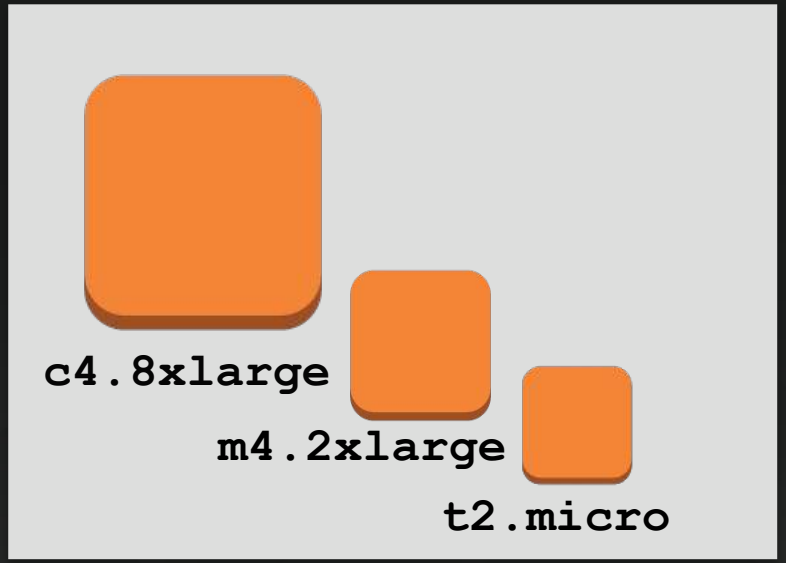
Day 1, user 1

- A single Amazon EC2 instance, with full stack on this host
  - Web app
  - Database
  - Management
  - And so on…
- A single Elastic IP address
- Amazon Route 53 for DNS



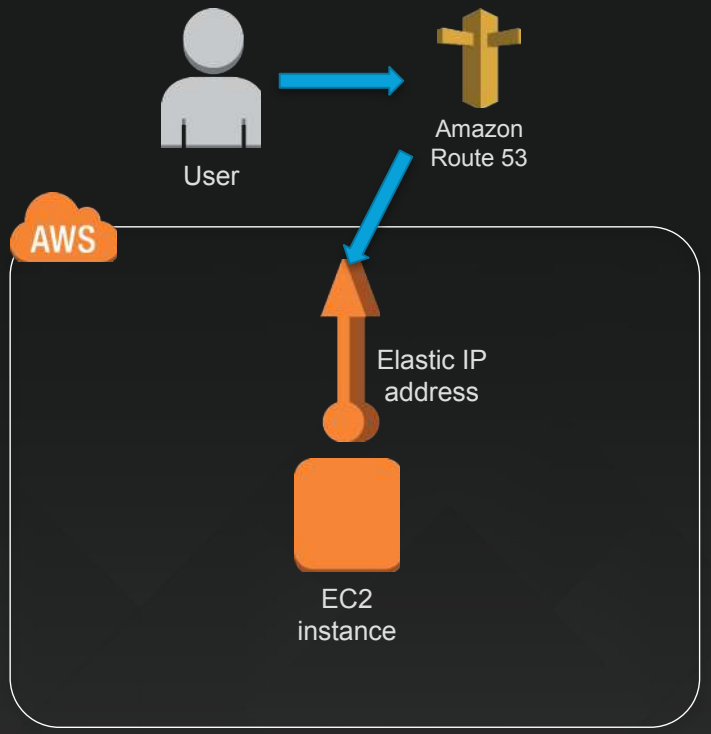
“We’re gonna need a bigger box”

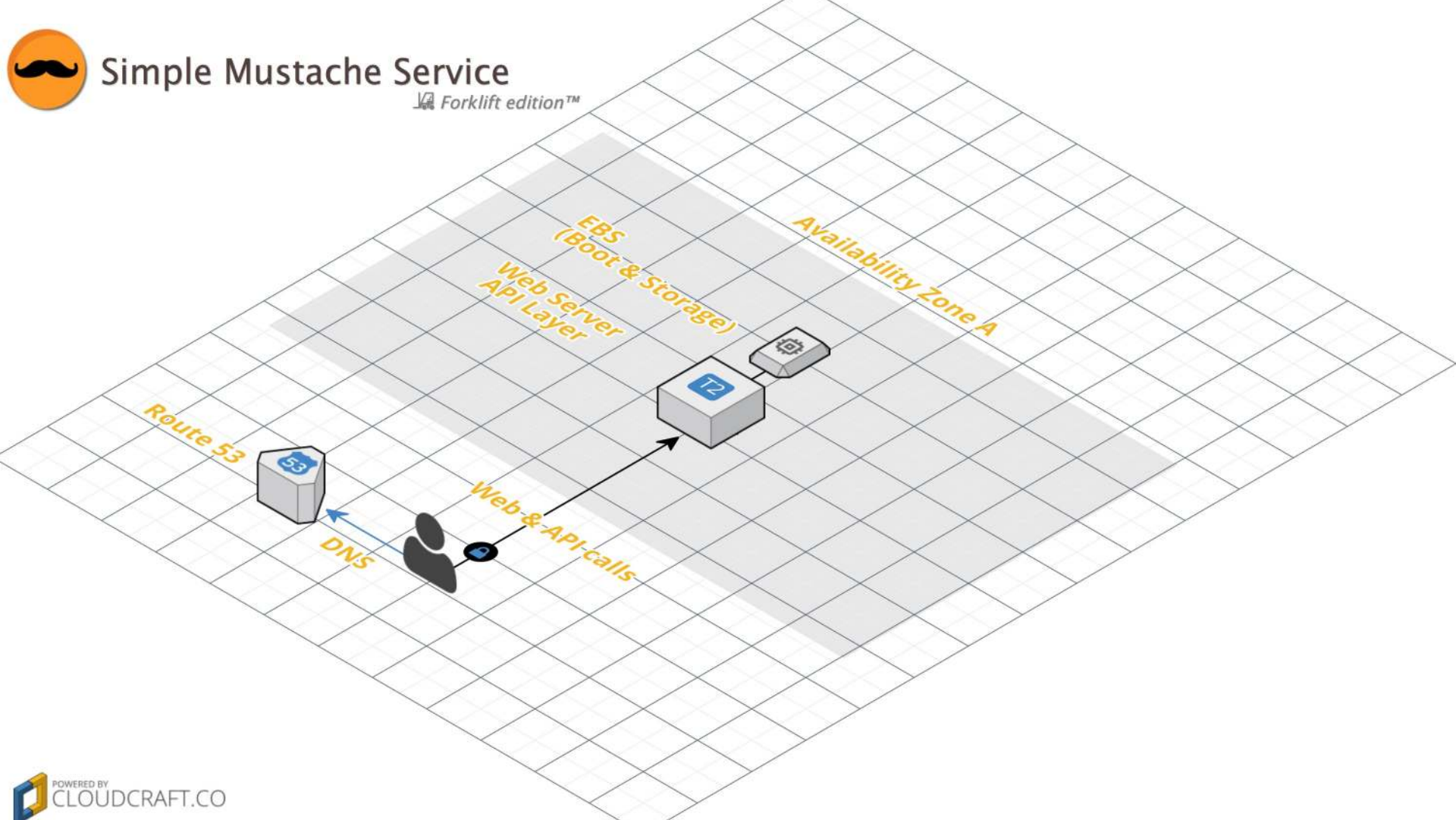
- “Scale up”: simplest approach
- Can now leverage PIOPS
- High I/O instances
- High memory instances
- High CPU instances
- High storage instances
- Easy to change instance sizes
- **Will hit a wall eventually**



Day 1, user 1

- We could potentially get to a few hundred to a few thousand depending on application complexity and traffic
- **No failover**
- **No redundancy**
- **Too many eggs in one basket**



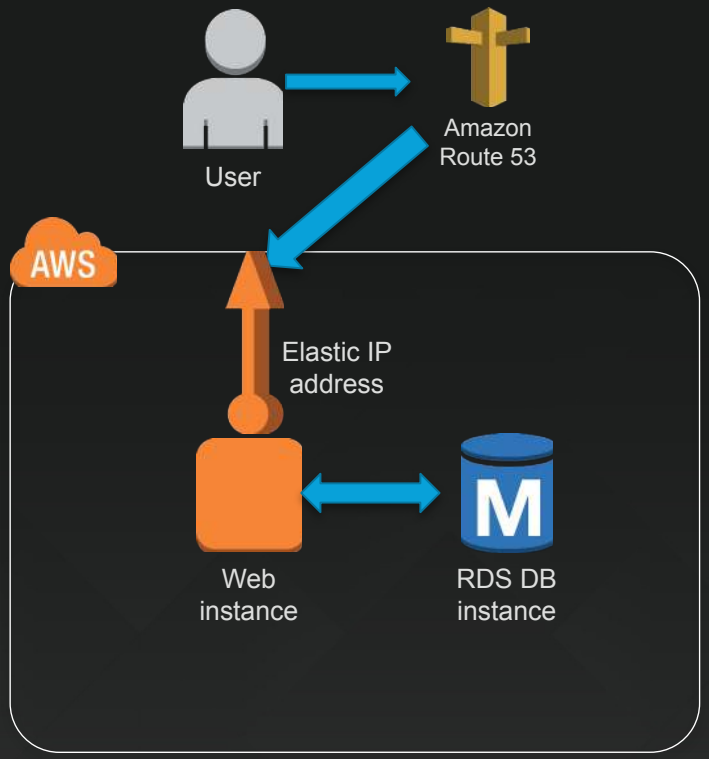

Simple Mustache Service
Forklift edition™
EBS
(Boot & Storage)
Web Server
API Layer
Availability Zone A
Route 53
S3
T2
Web-&-API-calls
DNS
POWERED BY
CLOUDCRAFT.CO

100 users

First, let's separate out our single host into more than one:

- Web
- Database

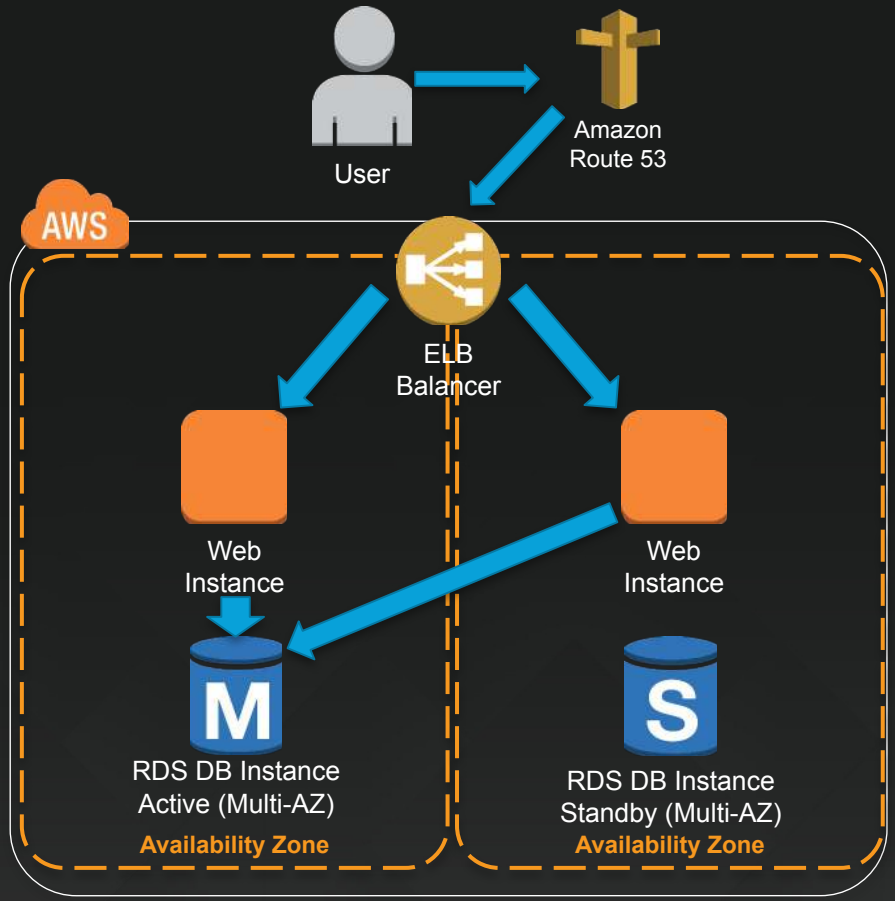
Use Amazon RDS to make your life easier



1000 users

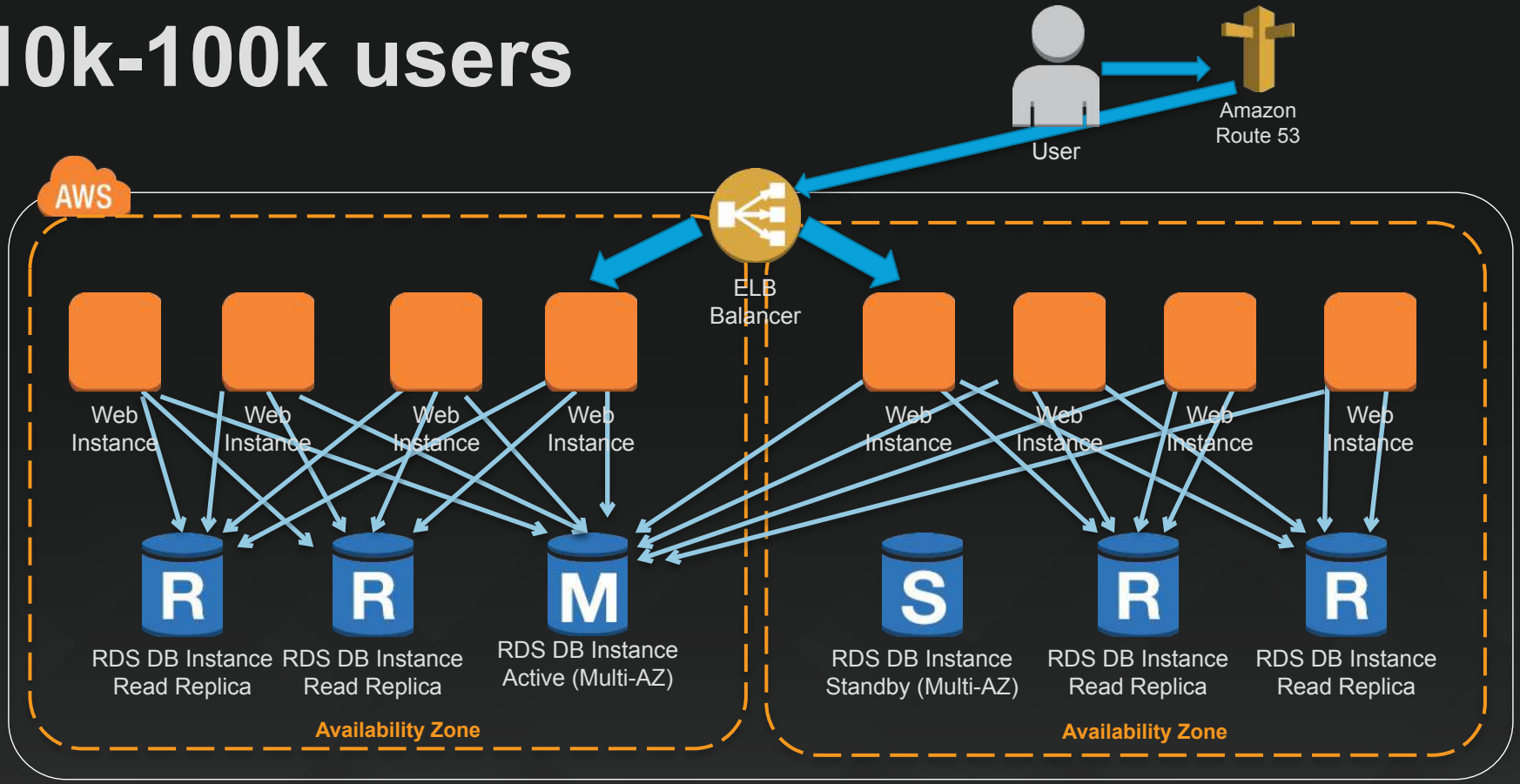
Next, let's address our lack of failover and redundancy issues:

- Elastic Load Balancing (ELB)
- Another web instance
  - In another Availability Zone
- RDS Multi-AZ



Scaling this horizontally and vertically will get us pretty far (tens to hundreds of thousands)

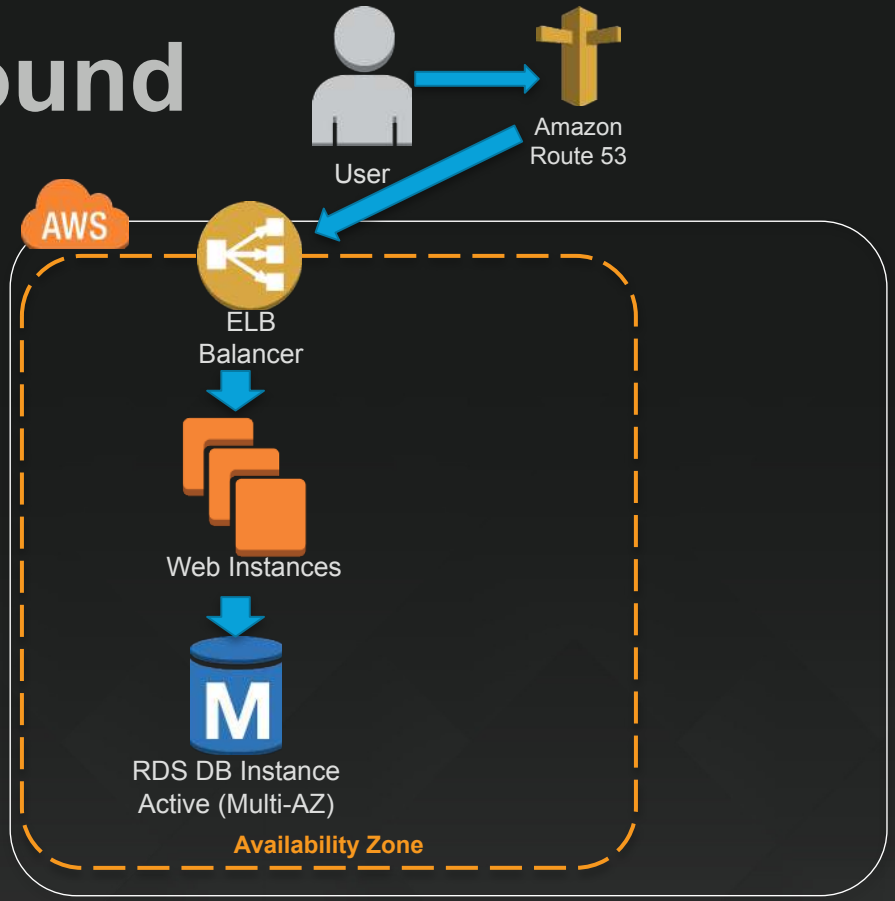
10k-100k users



This will take us pretty far, but we care about *performance* and *efficiency*, so let's improve further

Shift some load around

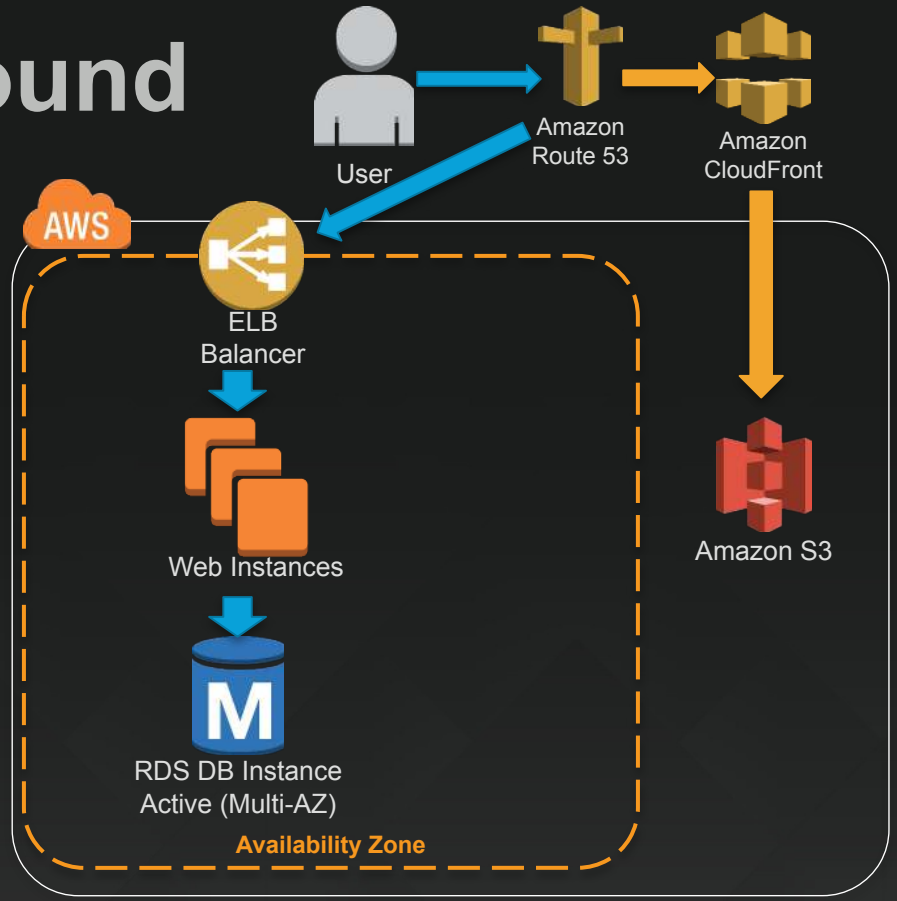
Let’s lighten the load on our web and database instances



Shift some load around

Let's lighten the load on our web and database instances:

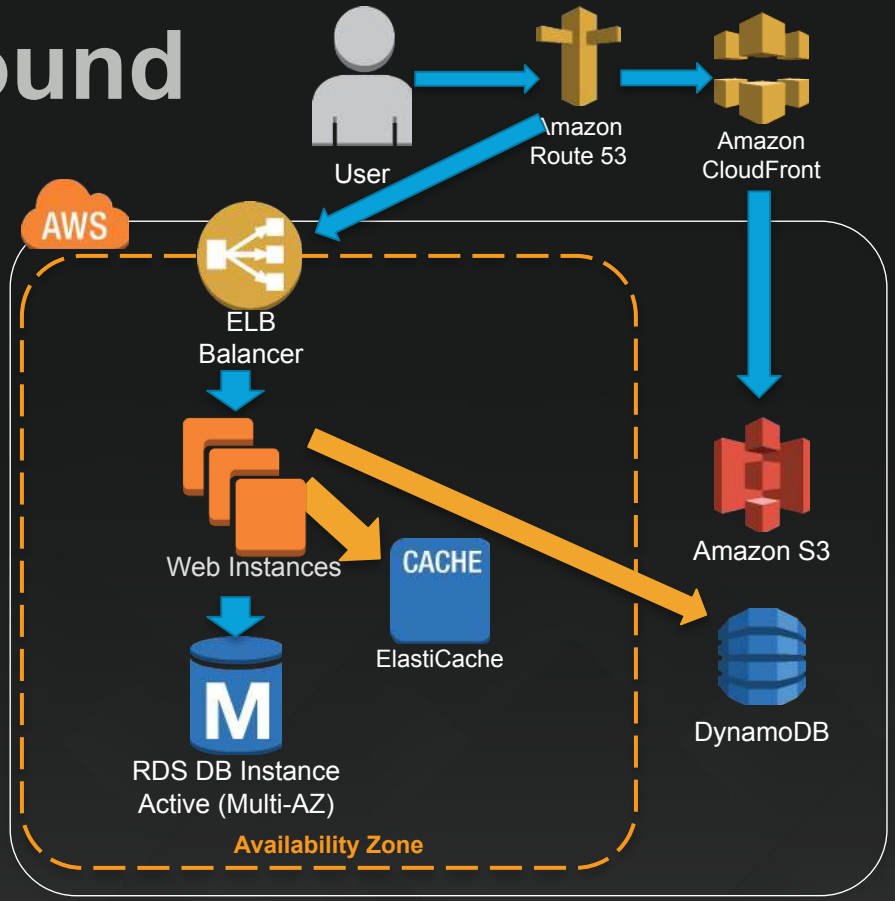
- **Move static content from the web instance to Amazon S3 and Amazon CloudFront**
- **We could even host the site directly in S3 (aka "S3 static website")**



Shift some load around

Let's lighten the load on our web and database instances:

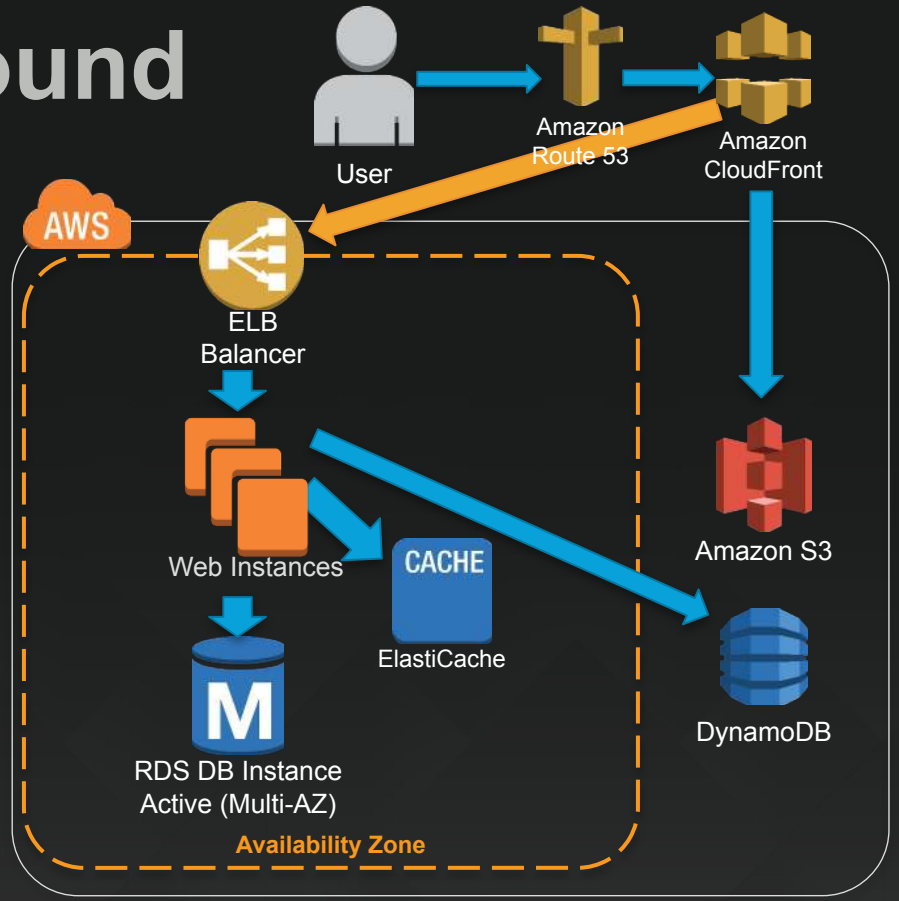
- Move static content from the web instance to Amazon S3 and Amazon CloudFront
- **Move session/state and DB caching to Amazon ElastiCache or Amazon DynamoDB**



Shift some load around

Let's lighten the load on our web and database instances:

- Move static content from the web instance to Amazon S3 and Amazon CloudFront
- Move session/state and DB caching to ElastiCache or DynamoDB
- **Move dynamic content from the ELB balancer to Amazon CloudFront**



amazon web services

Now that our web tier is much more lightweight…

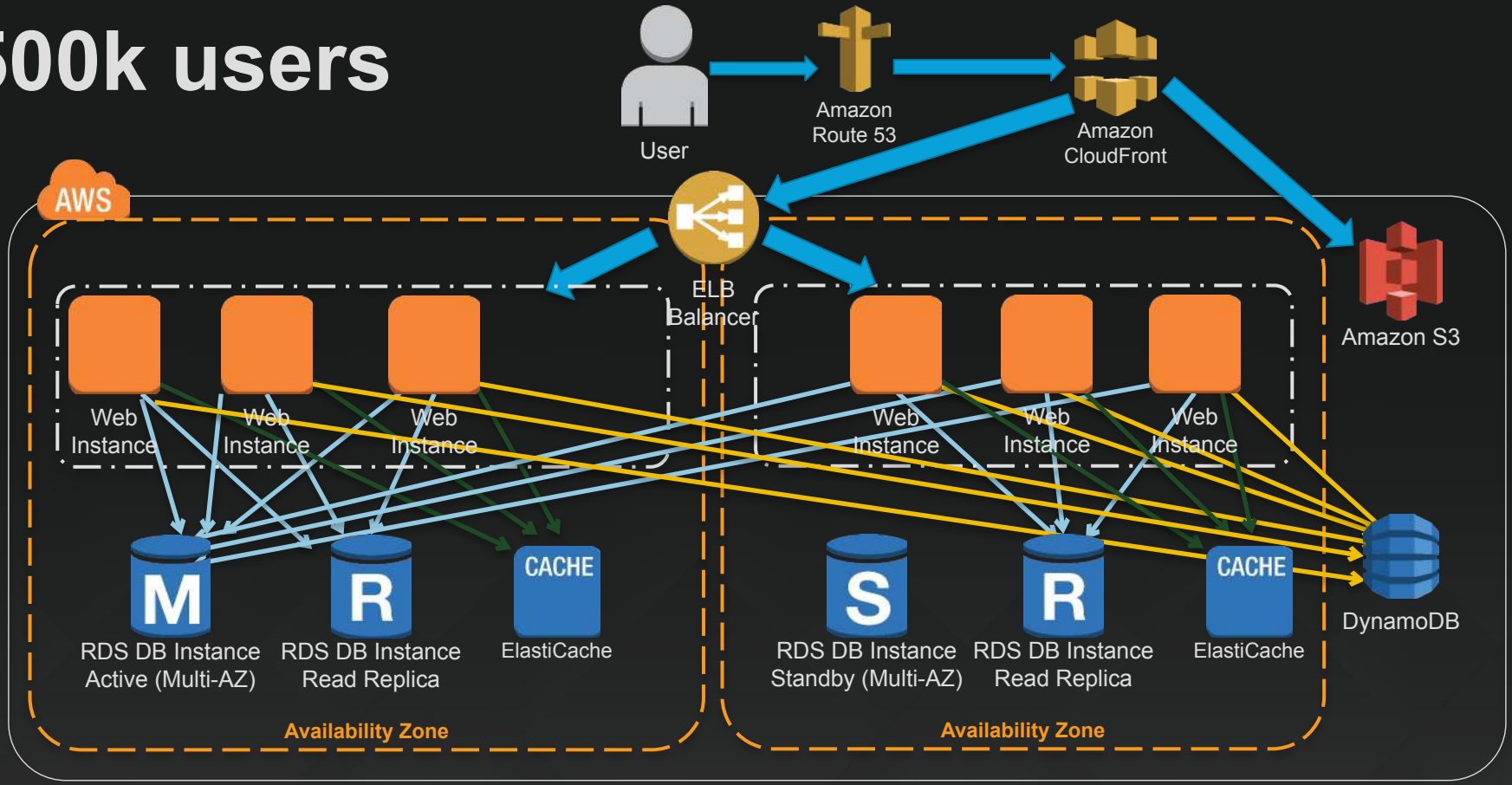
Auto Scaling!

Resize server farms automatically based on monitoring metrics

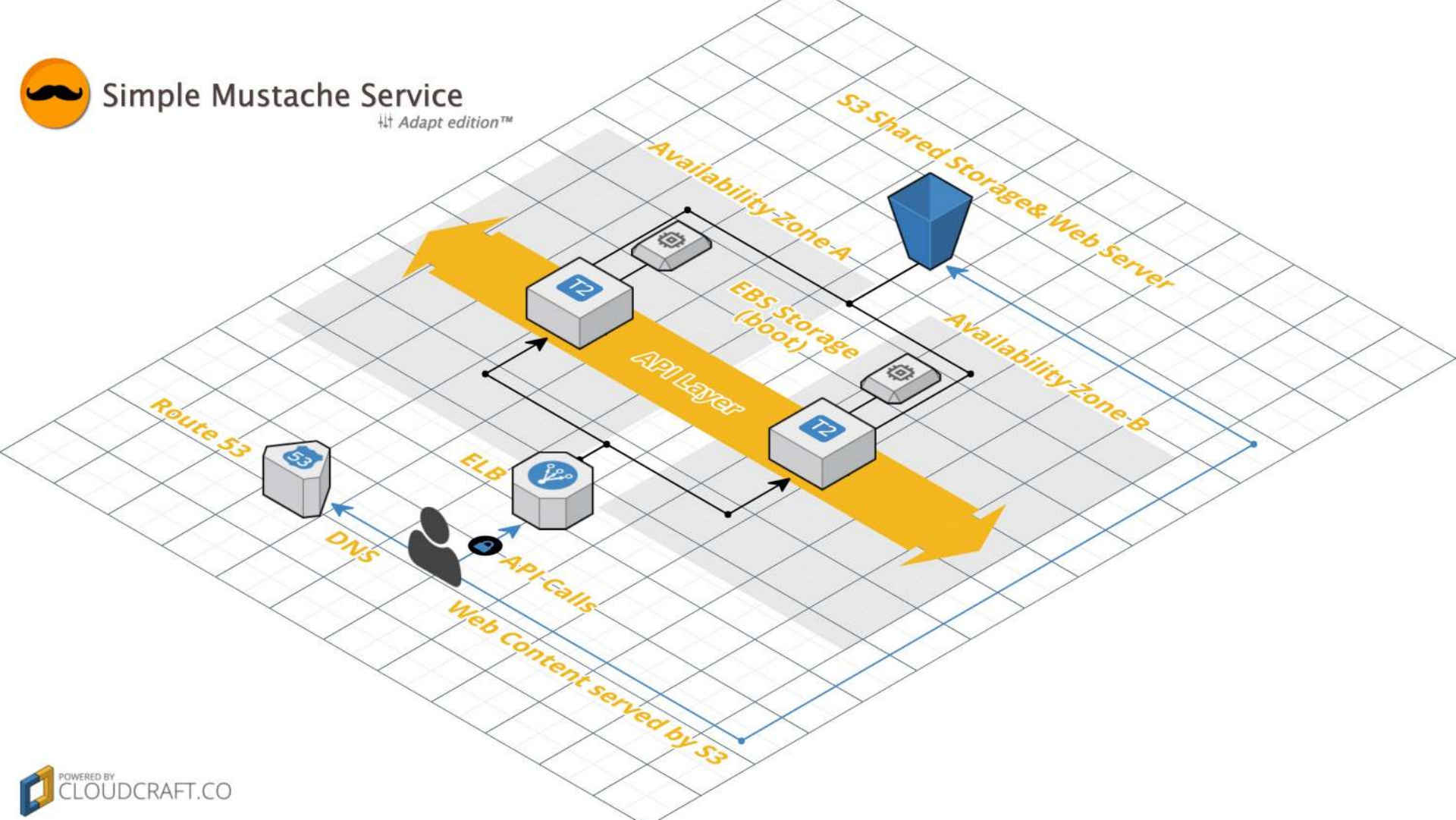
Spot instances!

Bid on unused EC2 capacity (typically at 80% discount)

500k users



amazon web services


Simple Mustache Service
Adapt edition™
S3 Shared Storage& Web Server
Availability Zone A
EBS Storage
(boot)
API Layer
Availability Zone B
T2
T2
Route 53
S3
ELB
DNS
API Calls
Web Content served by S3
POWERED BY
CLOUDCRAFT.CO

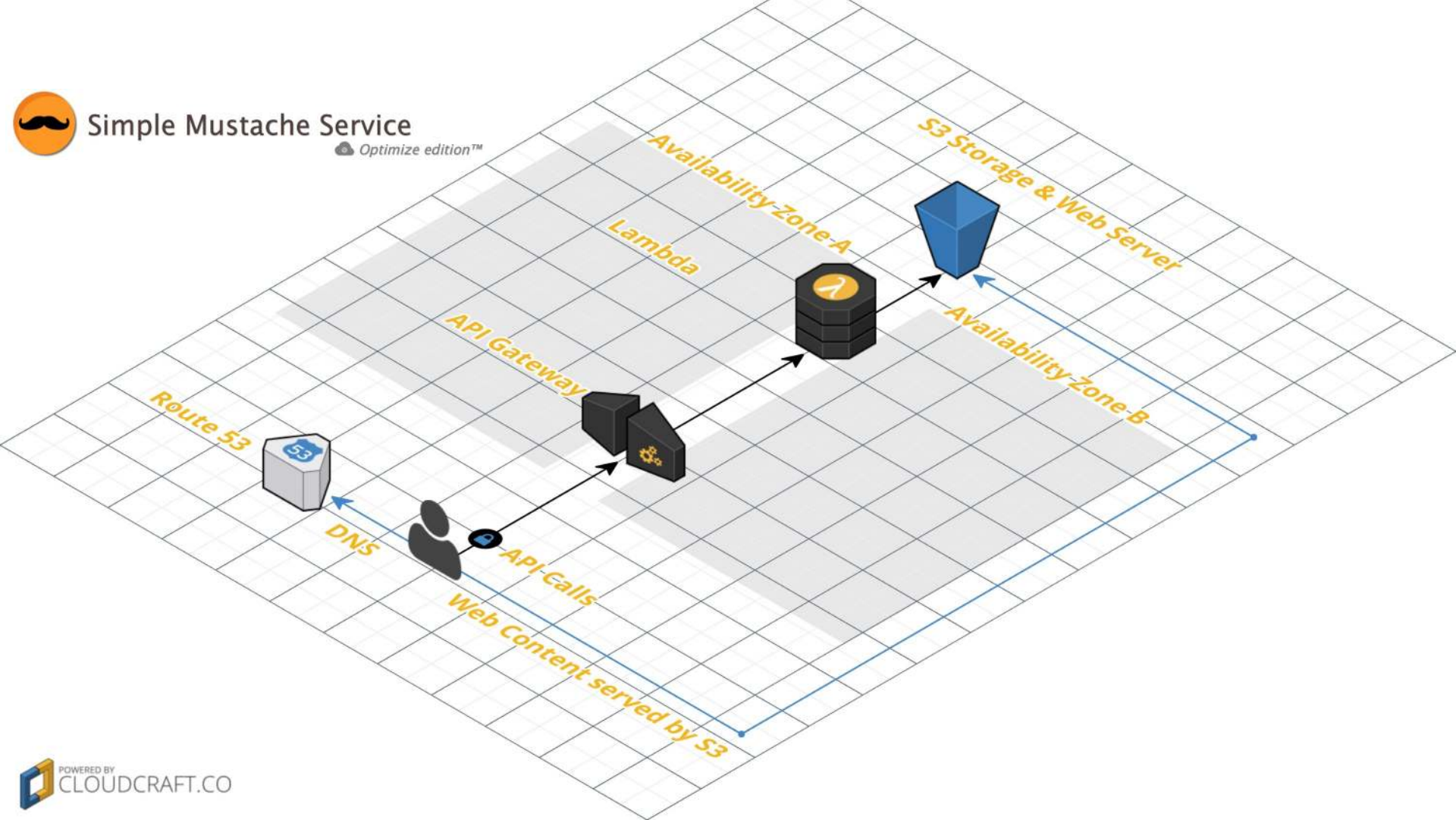
There are more improvements to be made and we could get much higher, but do we really want to manage all these instances ?

Werner Vogels, CTO, Amazon.com
AWS re:Invent 2015

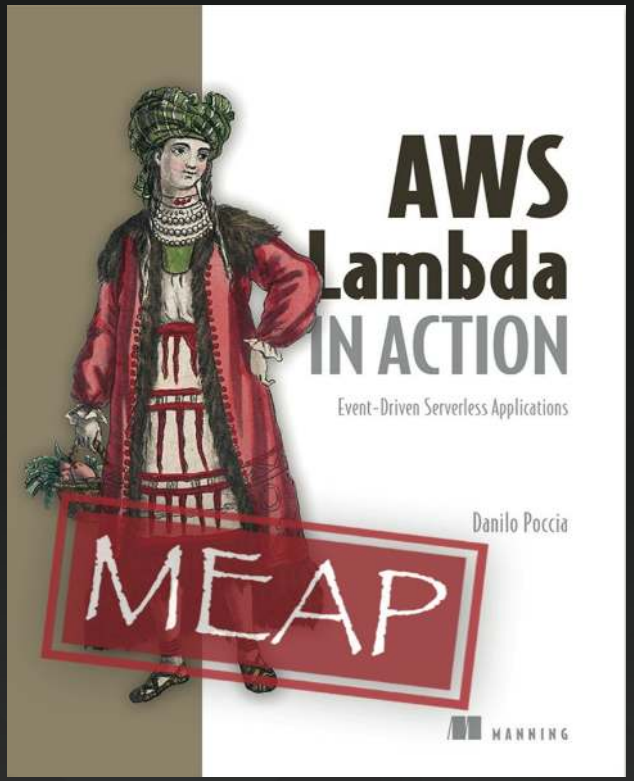
amazon web services

Use the Force, Luke!

Managed services
+ AWS Lambda
= Serverless architecture


Simple Mustache Service
Optimize edition™
Availability-Zone-A
Lambda
API Gateway
Route 53
S3 Storage & Web Server
Availability-Zone-B
DNS
API Calls
Web Content served by S3
POWERED BY
CLOUDCRAFT.CO

Upcoming book on AWS Lambda



Written by AWS Technical Evangelist Danilo Poccia

Early release available at:

https://www.manning.com/books/aws-lambda-in-action

amazon web services

Thank You !

Julien Simon
julsimon@amazon.fr
@julsimon