# Classic Solutions Architecture

Section Introduction

- These solutions architectures are the best part of this course

- Let's understand how all the technologies we've seen work together

- This is a section you need to be 100% comfortable with

- We'll see the progression of a Solution's architect mindset through many sample case studies:
  - WhatIsTheTime.Com
  - MyClothes.Com
  - MyWordPress.Com
  - Instantiating applications quickly
  - Beanstalk

Stateless Web App: WhatIsTheTime.com

- WhatIsTheTime.com allows people to know what time it is
- We don't need a database
- We want to start small and can accept downtime
- We want to fully scale vertically and horizontally, no downtime
- Let's go through the Solutions Architect journey for this app
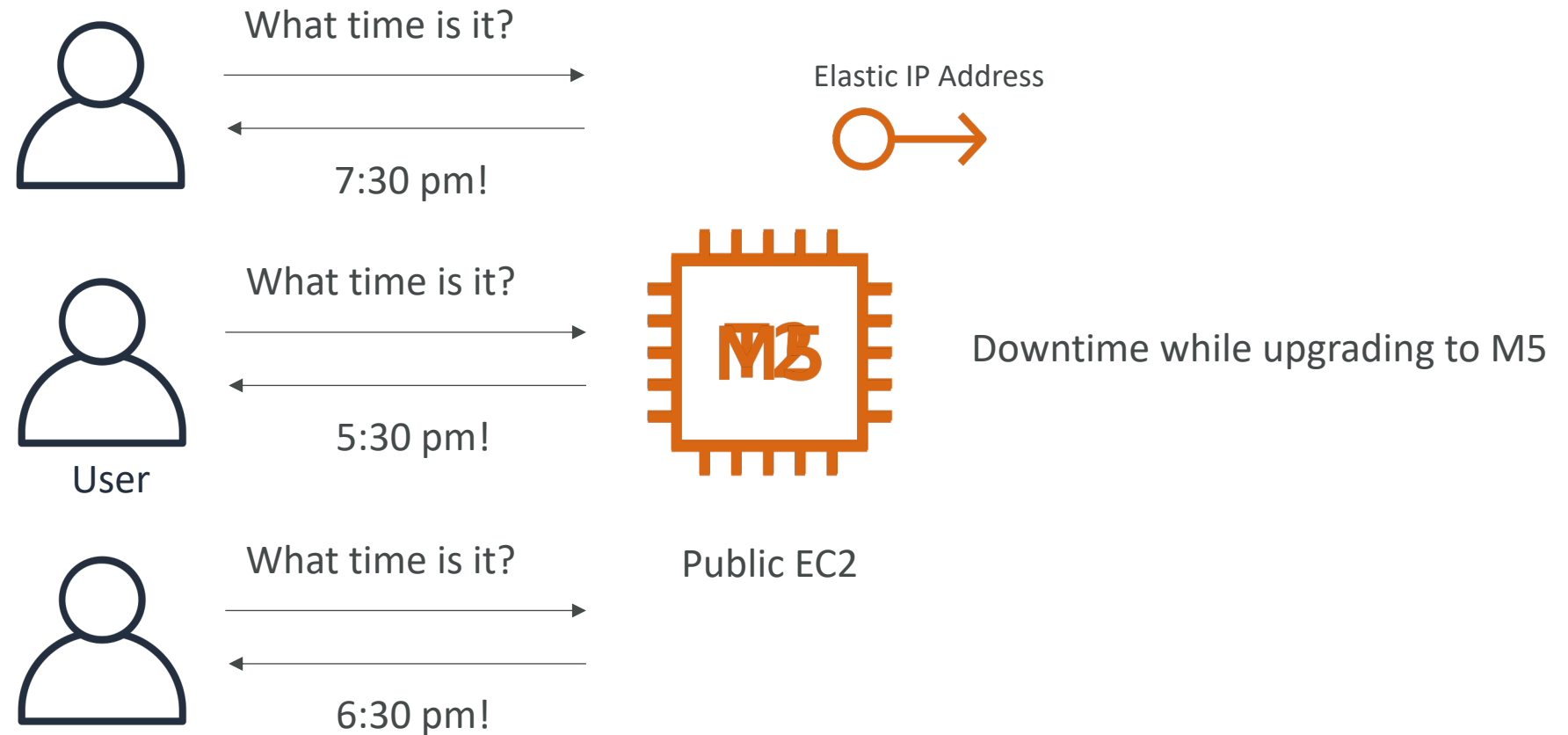
- Let's see how we can proceed!

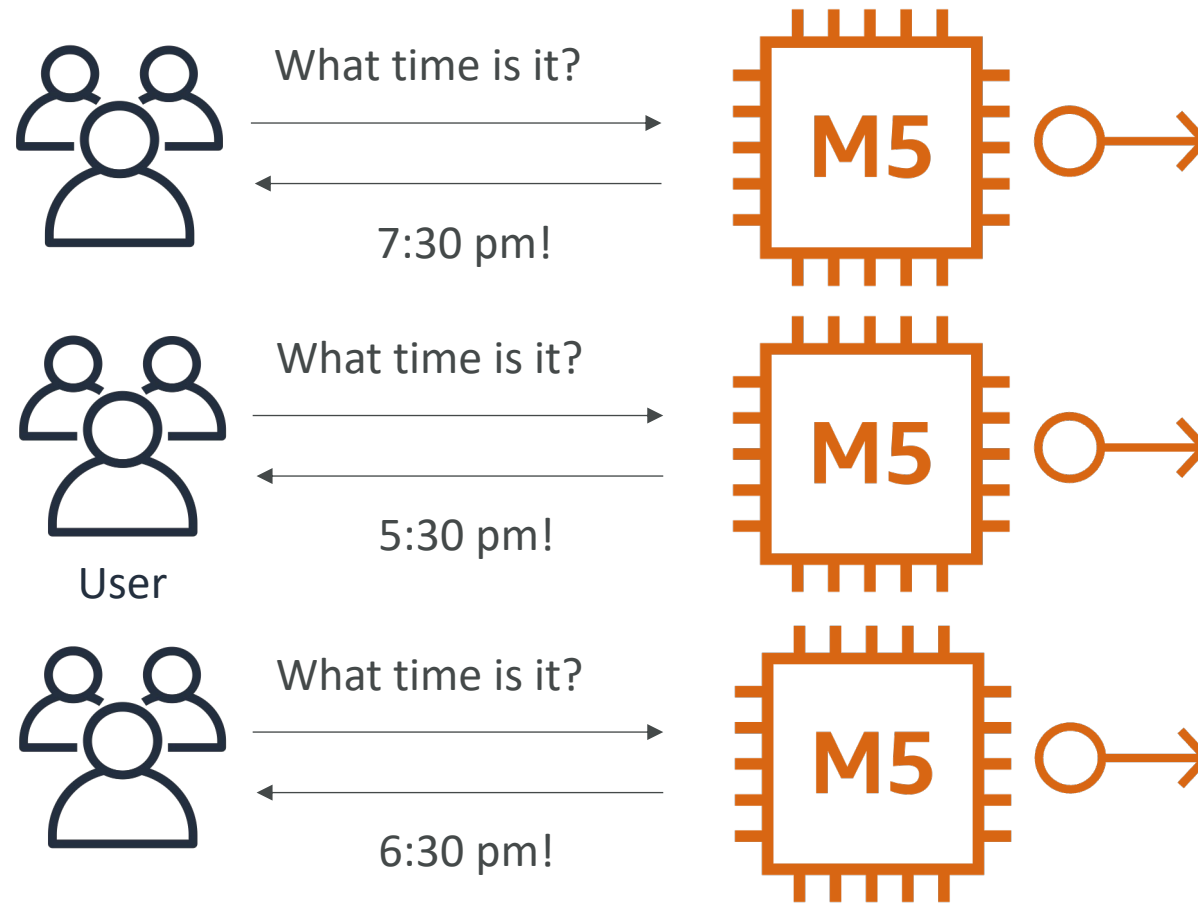# Stateless web app: What time is it? Starting simple
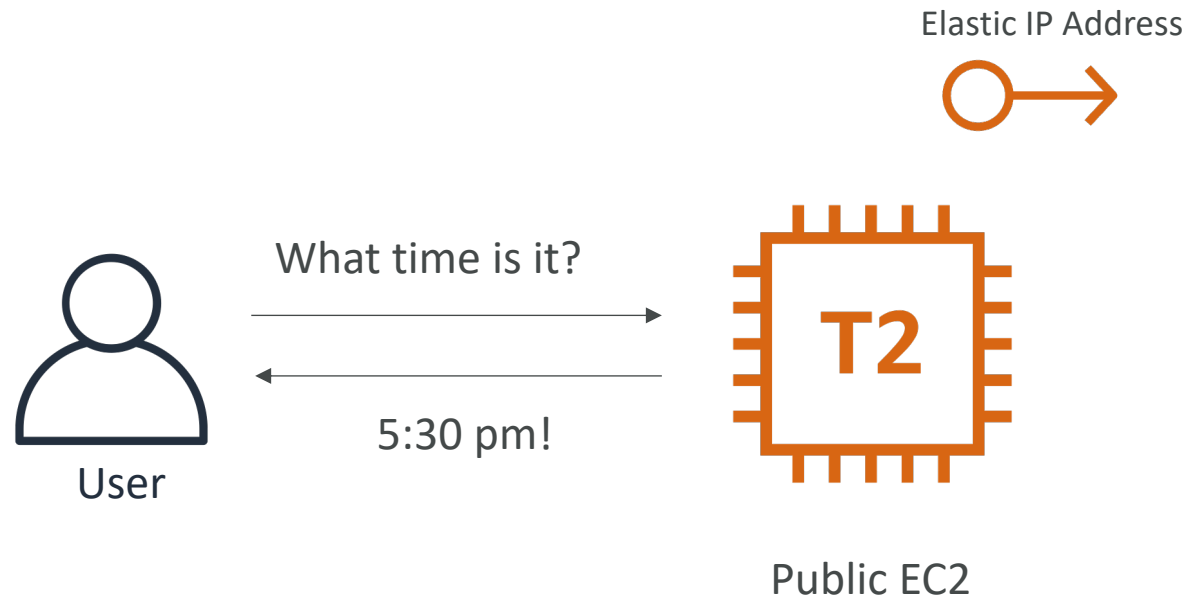
Stateless web app: What time is it?
Scaling vertically

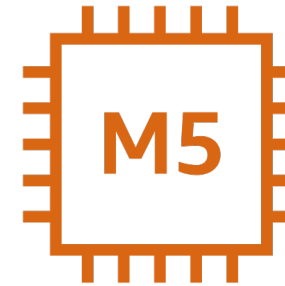# Stateless web app: What time is it? Scaling horizontally

What time is it?

7:30 pm!

What time is it?

5:30 pm!

User

What time is it?

6:30 pm!

M5

M5

M5

Stateless web app: What time is it? Scaling horizontally

DNS Query
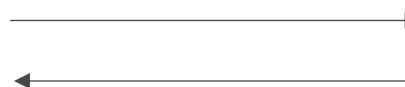For api.whatisthetime.com
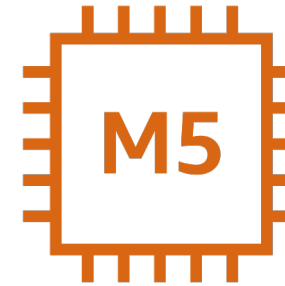A Record
TTL 1 hour

53

What time is it?

7:30 pm!

What time is it?

5:30 pm!

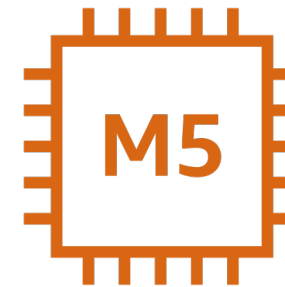What time is it?

6:30 pm!

M5

M5

M5

Public EC2 instance,
No Elastic IP

Stateless web app: What time is it?
Scaling horizontally, adding and removing instances

DNS Query
For api.whatisthetime.com
A Record
TTL 1 hour

53

What time is it?

7:30 pm!

INSTANCE IS GONE!
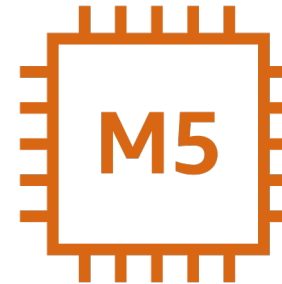
What time is it?

5:30 pm!

M5

Public EC2 instance,
No Elastic IP

What time is it?

6:30 pm!

M5
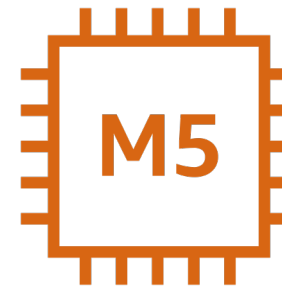
Stateless web app: What time is it?
Scaling horizontally, with a load balancer

What time is it?

Availability zone 1

Restricted
Security groups rules

Availability zone 1

M5

M5

M5

ELB +
Health Checks

DNS Query
For api.whatisthetime.com
Alias Record

Private
EC2 instances
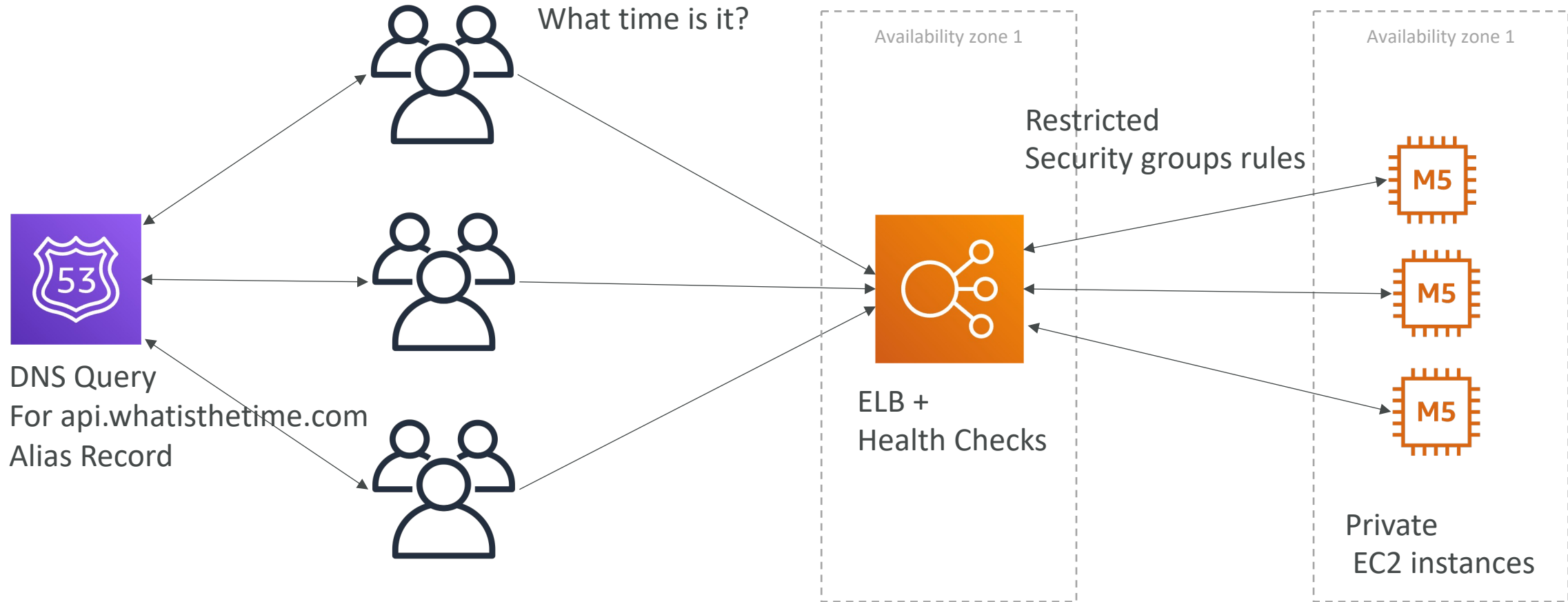
Stateless web app: What time is it?
Scaling horizontally, with an auto-scaling group

What time is it?

DNS Query
For api.whatisthetime.com
Alias Record

Availability zone 1

Availability zone 1

Auto Scaling group

M5

M5

M5

ELB +
Health Checks

Private
EC2 instances
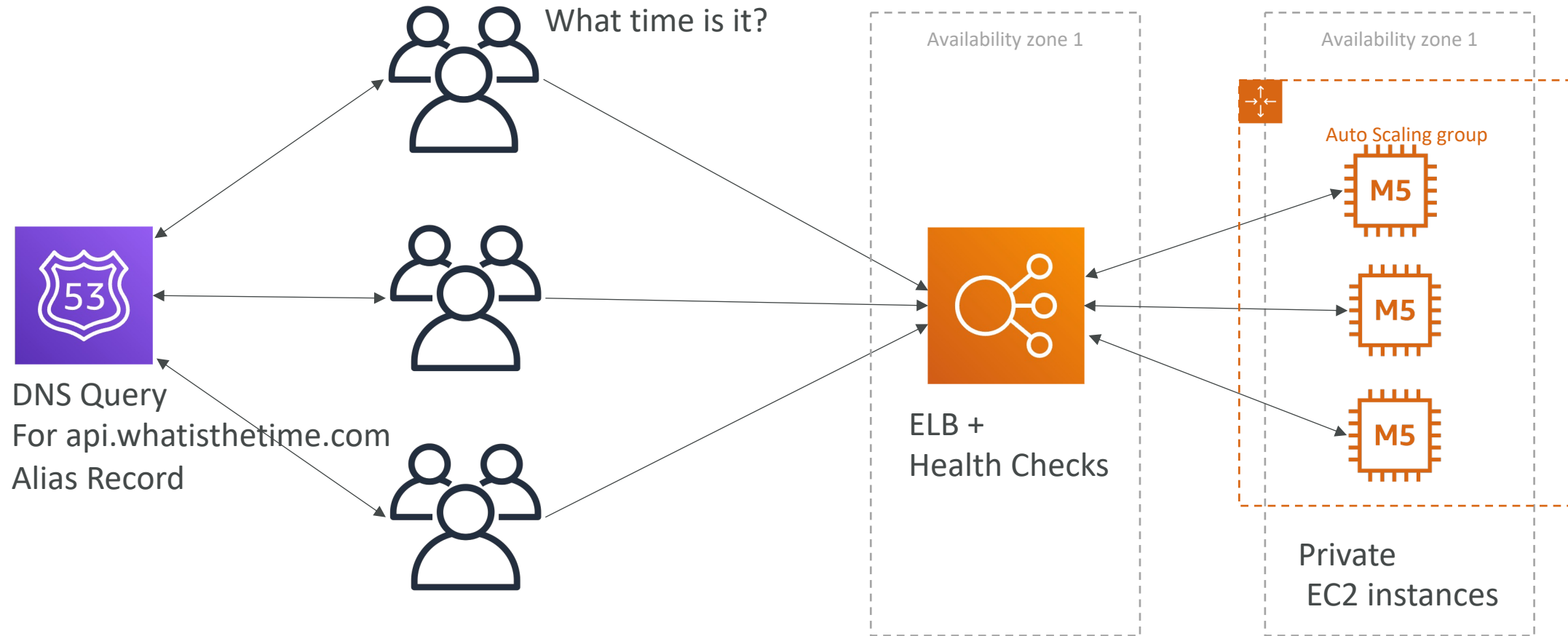
Stateless web app: What time is it? Making our app multi-AZ

What time is it?

Auto Scaling group

Availability zone 1 to 3

Availability zone 1

M5

M5

DNS Query
For api.whatisthetime.com
Alias Record

53

ELB +
Health Checks
+ Multi AZ

Availability zone 2

M5

M5

Availability zone 3

M5

Minimum 2 AZ => Let's reserve capacity

Auto Scaling group

Availability zone 1 to 3

Availability zone 1

M5

Availability zone 2

M5

DNS Query
For api.whatisthetime.com
Alias Record

ELB +
Health Checks
+ Multi AZ

Minimum capacity
= reserved instances
= cost savings
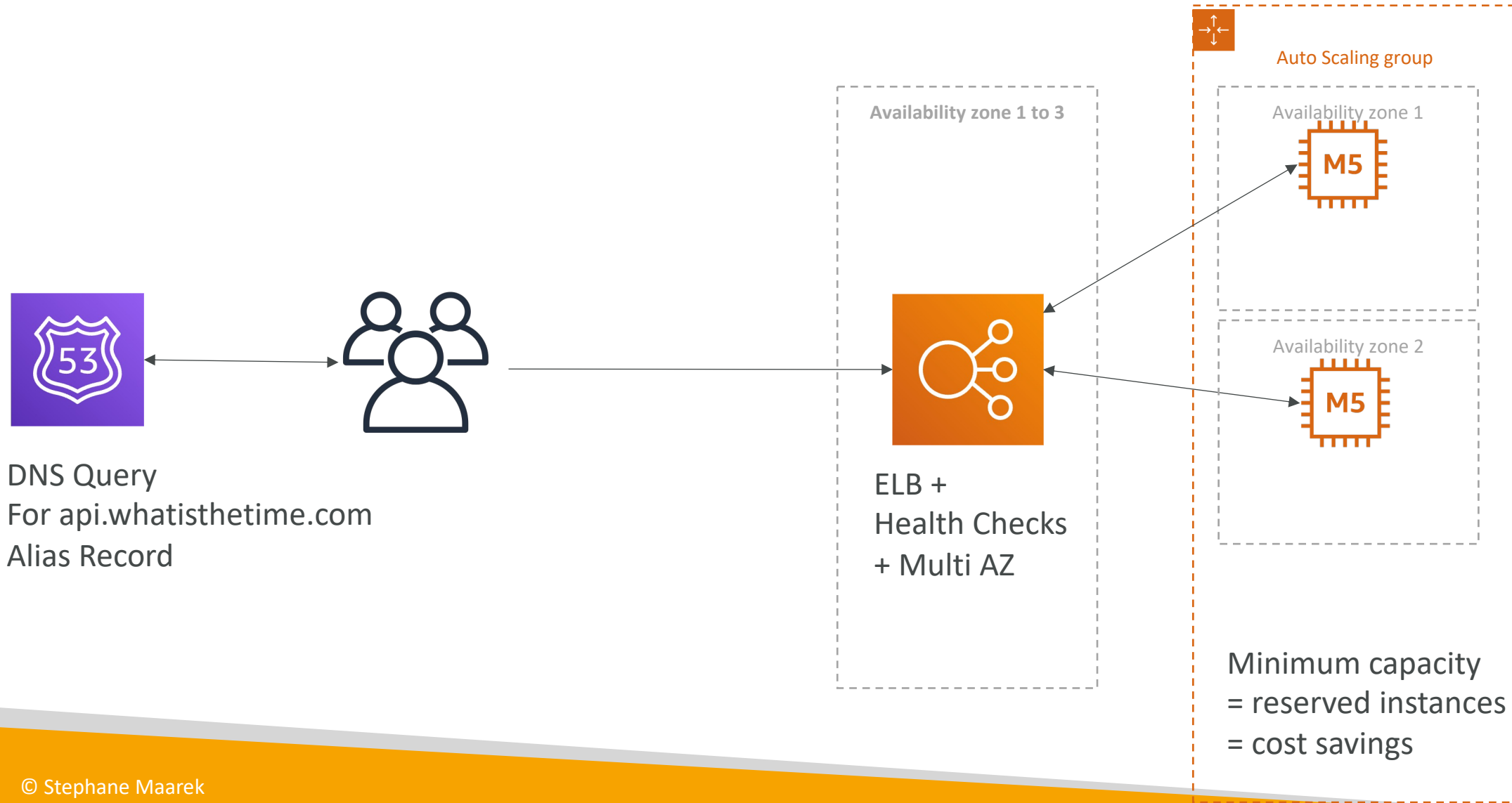
In this lecture we've discussed…

- Public vs Private IP and EC2 instances
- Elastic IP vs Route 53 vs Load Balancers
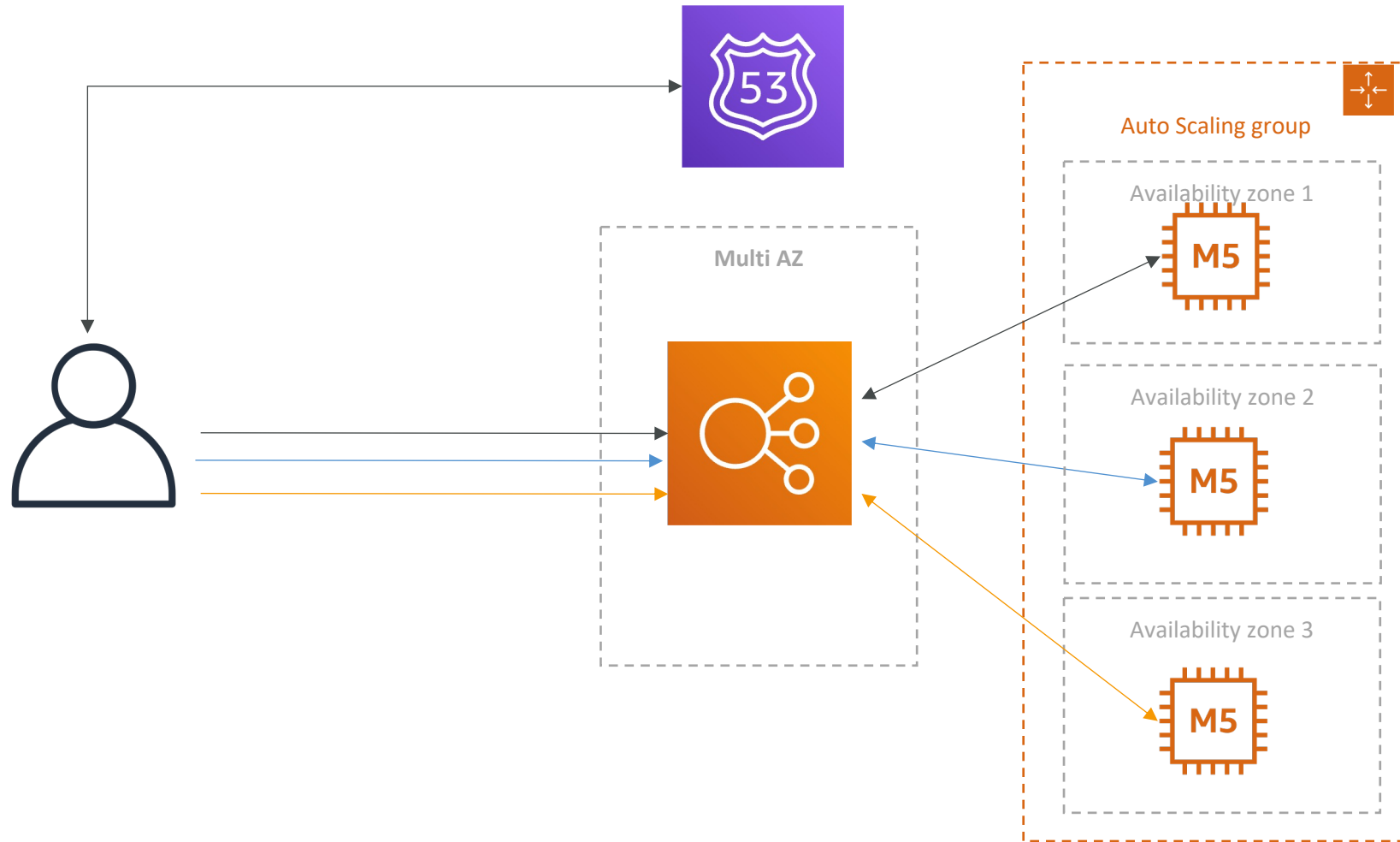- Route 53 TTL, A records and Alias Records
- Maintaining EC2 instances manually vs Auto Scaling Groups
- Multi AZ to survive disasters
- ELB Health Checks
- Security Group Rules
- Reservation of capacity for costing savings when possible

Stateful Web App: MyClothes.com

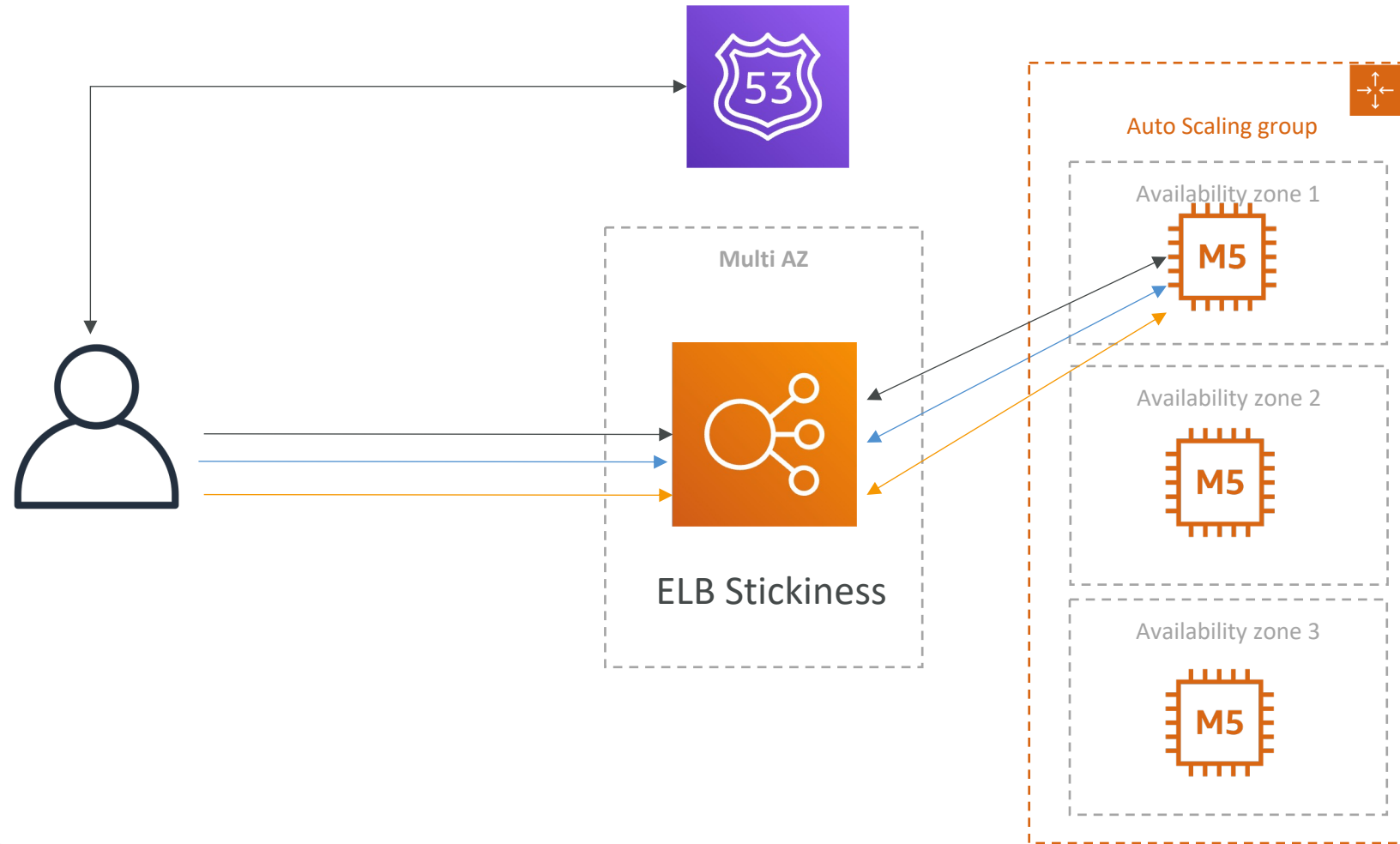- MyClothes.com allows people to buy clothes online.

- There's a shopping cart

- Our website is having hundreds of users at the same time

- We need to scale, maintain horizontal scalability and keep our web application as stateless as possible

- Users should not lose their shopping cart

- Users should have their details (address, etc) in a database

- Let's see how we can proceed!
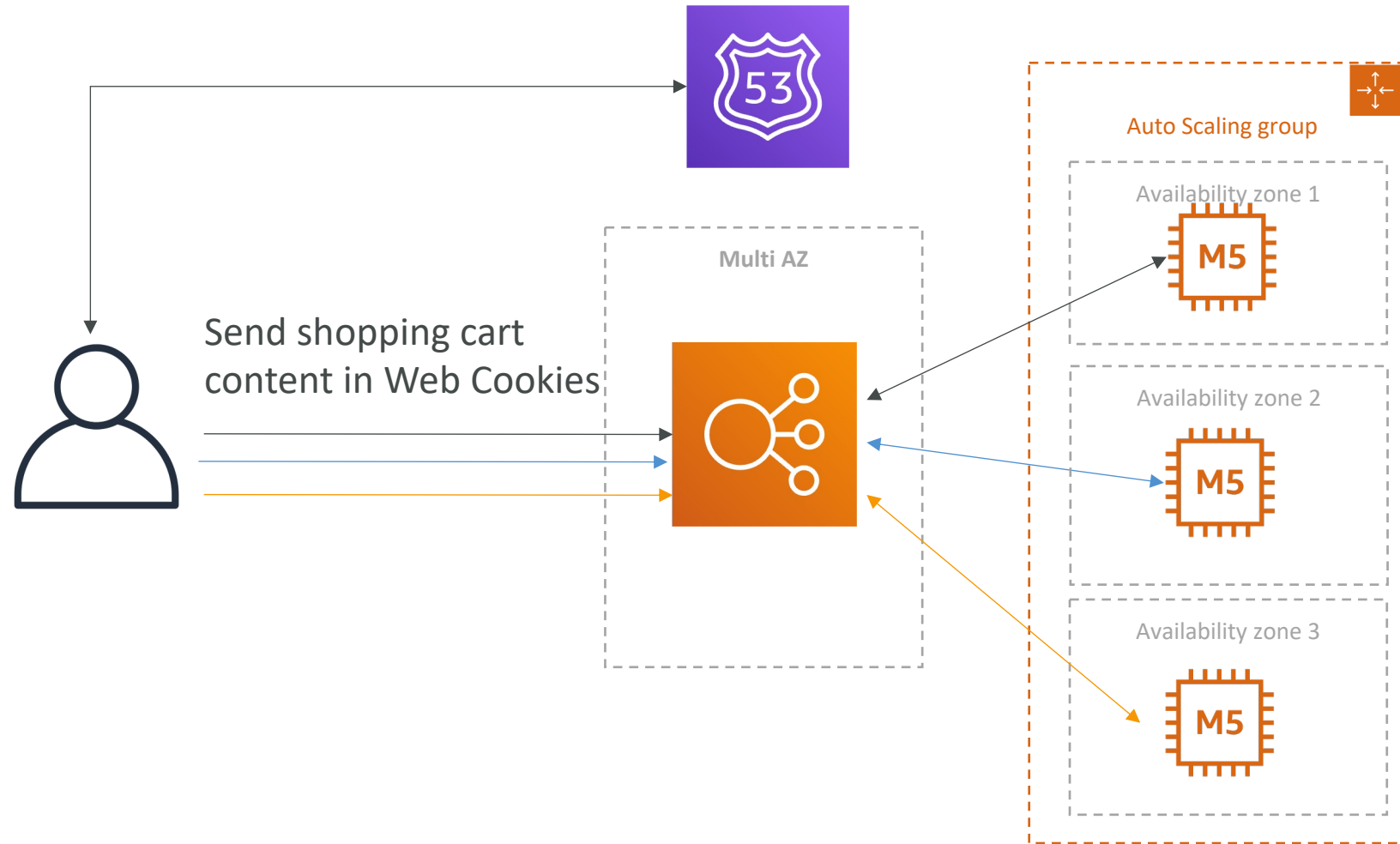
Stateful Web App: MyClothes.com

Stateful Web App: MyClothes.com
Introduce Stickiness (Session Affinity)

Auto Scaling group

Availability zone 1

M5

Multi AZ

Availability zone 2

M5

ELB Stickiness

Availability zone 3

M5

Stateful Web App: MyClothes.com Introduce User Cookies

Send shopping cart content in Web Cookies

Multi AZ

Auto Scaling group

Availability zone 1

M5

Availability zone 2

M5

Availability zone 3
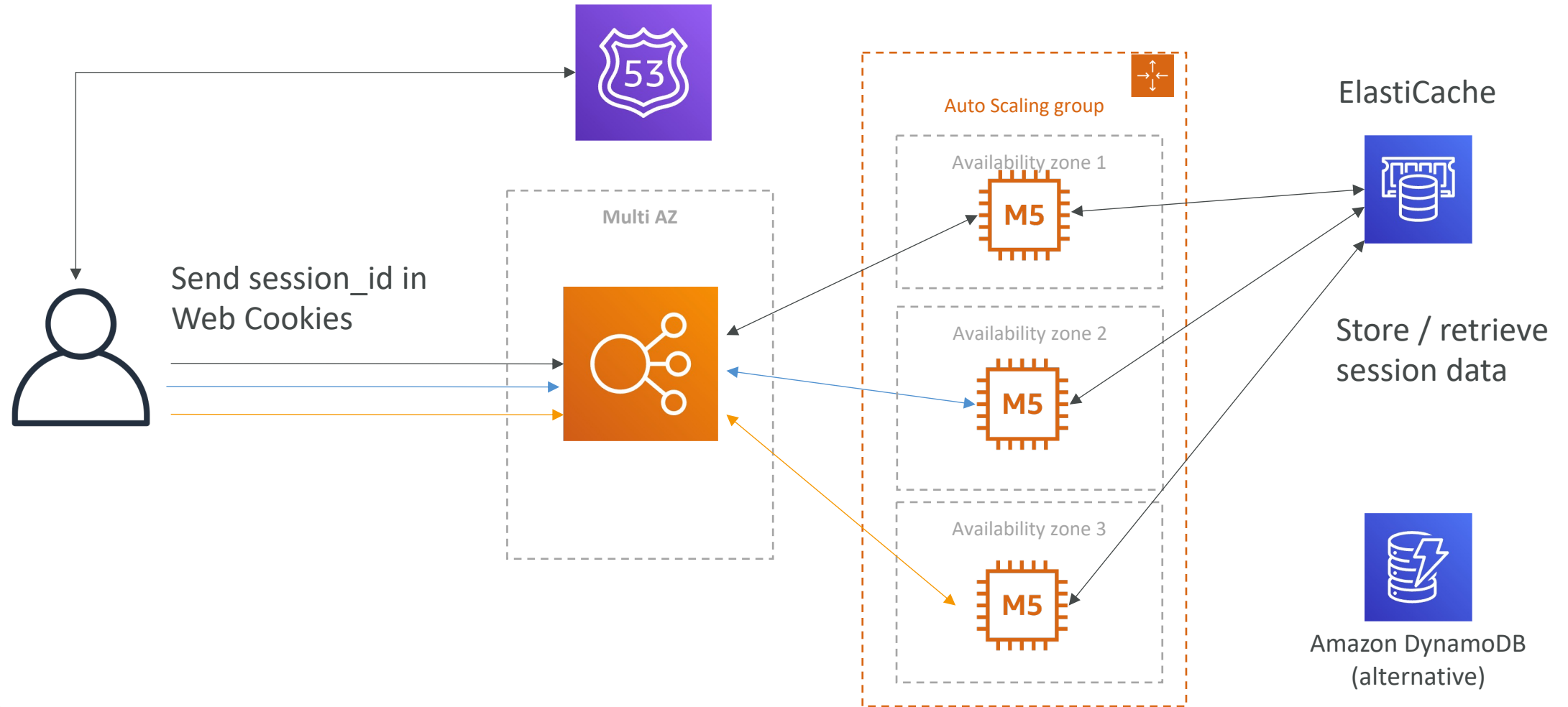
M5

Stateless
HTTP requests are heavier
Security risk
(cookies can be altered)
Cookies must be validated
Cookies must be less than 4KB

Stateful Web App: MyClothes.com
Introduce Server Session

ElastiCache

Store / retrieve
session data

Auto Scaling group

Availability zone 1

M5

Availability zone 2

M5

Availability zone 3

M5

Multi AZ

Send session_id in
Web Cookies

Amazon DynamoDB
(alternative)

Stateful Web App: MyClothes.com
Storing User Data in a database

© Stephane Maarek

Stateful Web App: MyClothes.com
Scaling Reads

© Stephane Maarek

www.datacumulus.com
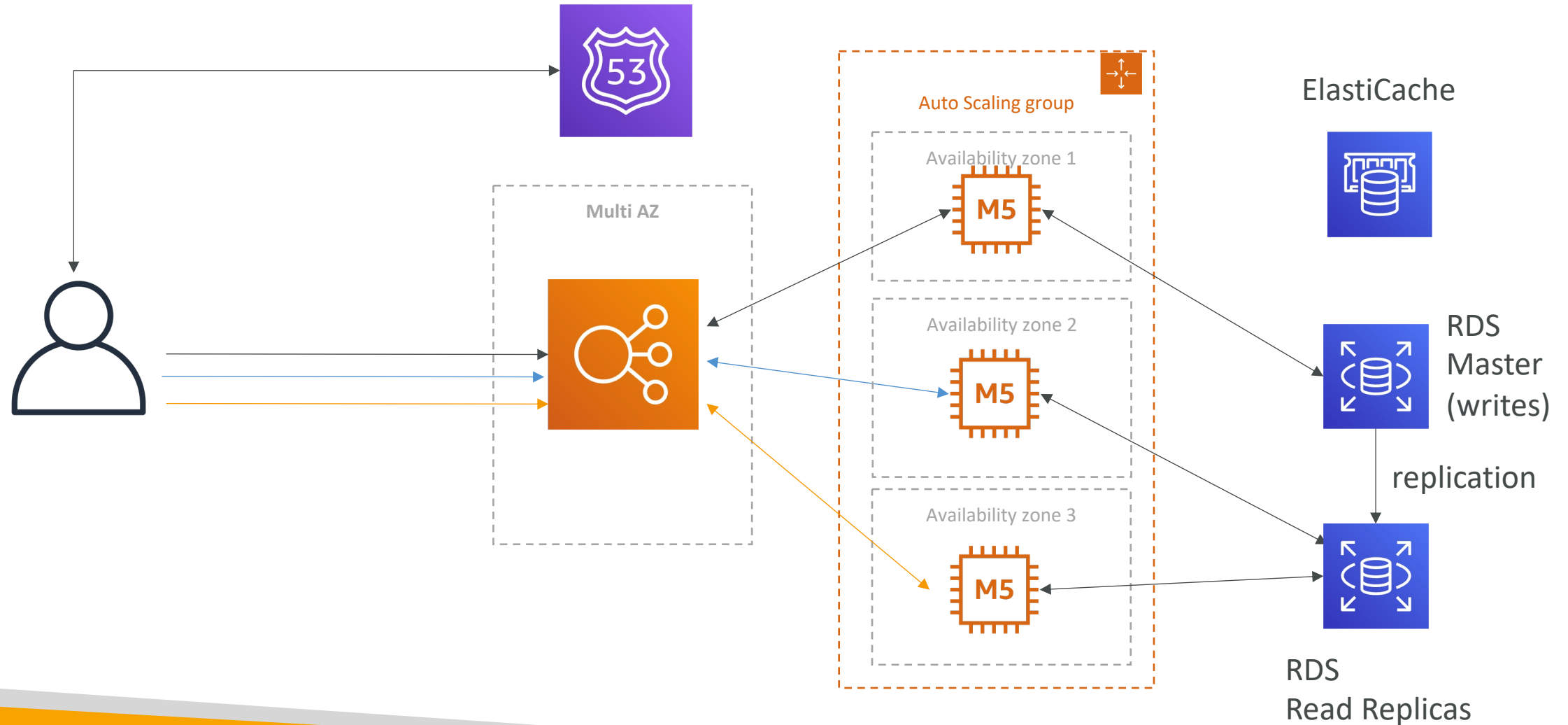
Stateful Web App: MyClothes.com
Scaling Reads (Alternative) – Lazy Loading

Auto Scaling group

ElastiCache

cache

hit

Read from cache

Multi AZ

Availability zone 1

M5

Availability zone 2

M5

Availability zone 3

M5

Read/write

RDS
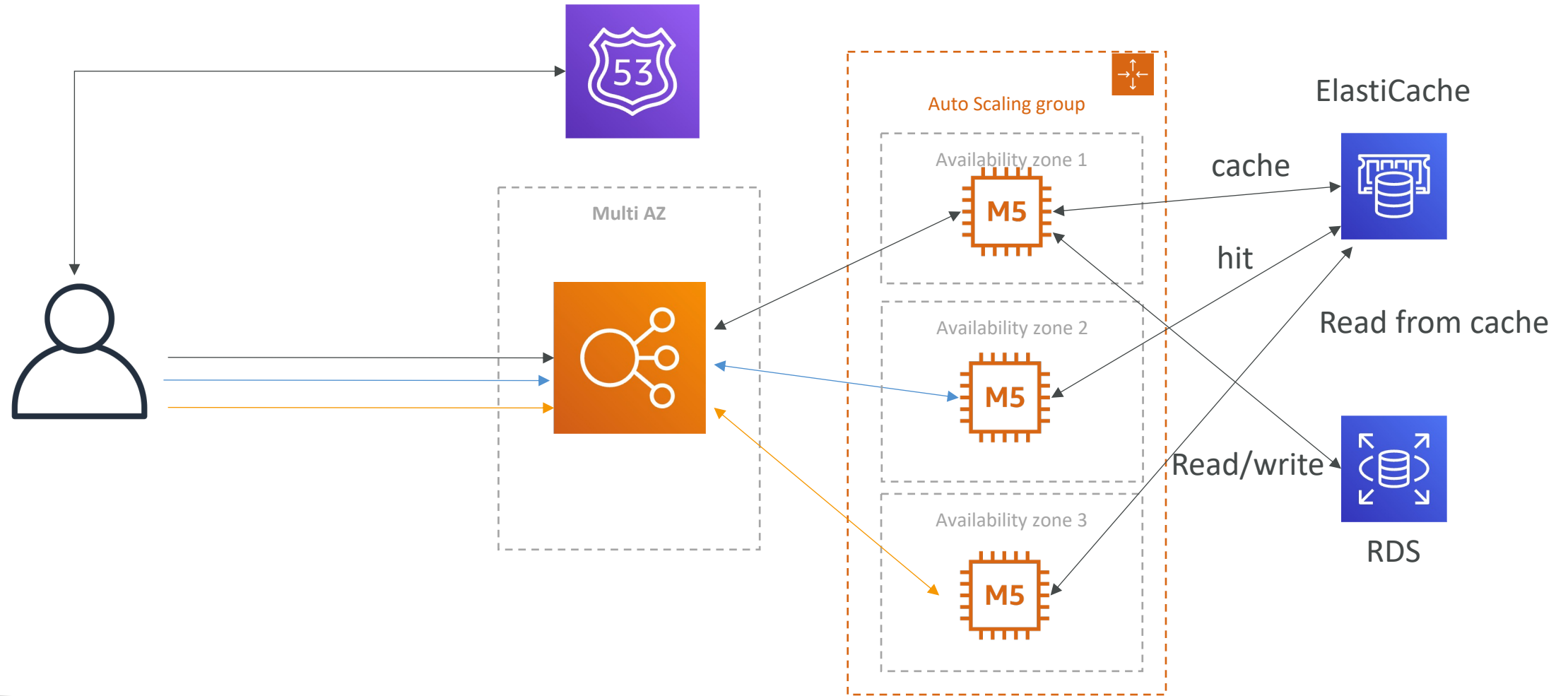
Stateful Web App: MyClothes.com
Multi AZ – Survive disasters

Multi AZ

Auto Scaling group

Availability zone 1

M5

Availability zone 2

M5

Availability zone 3

M5
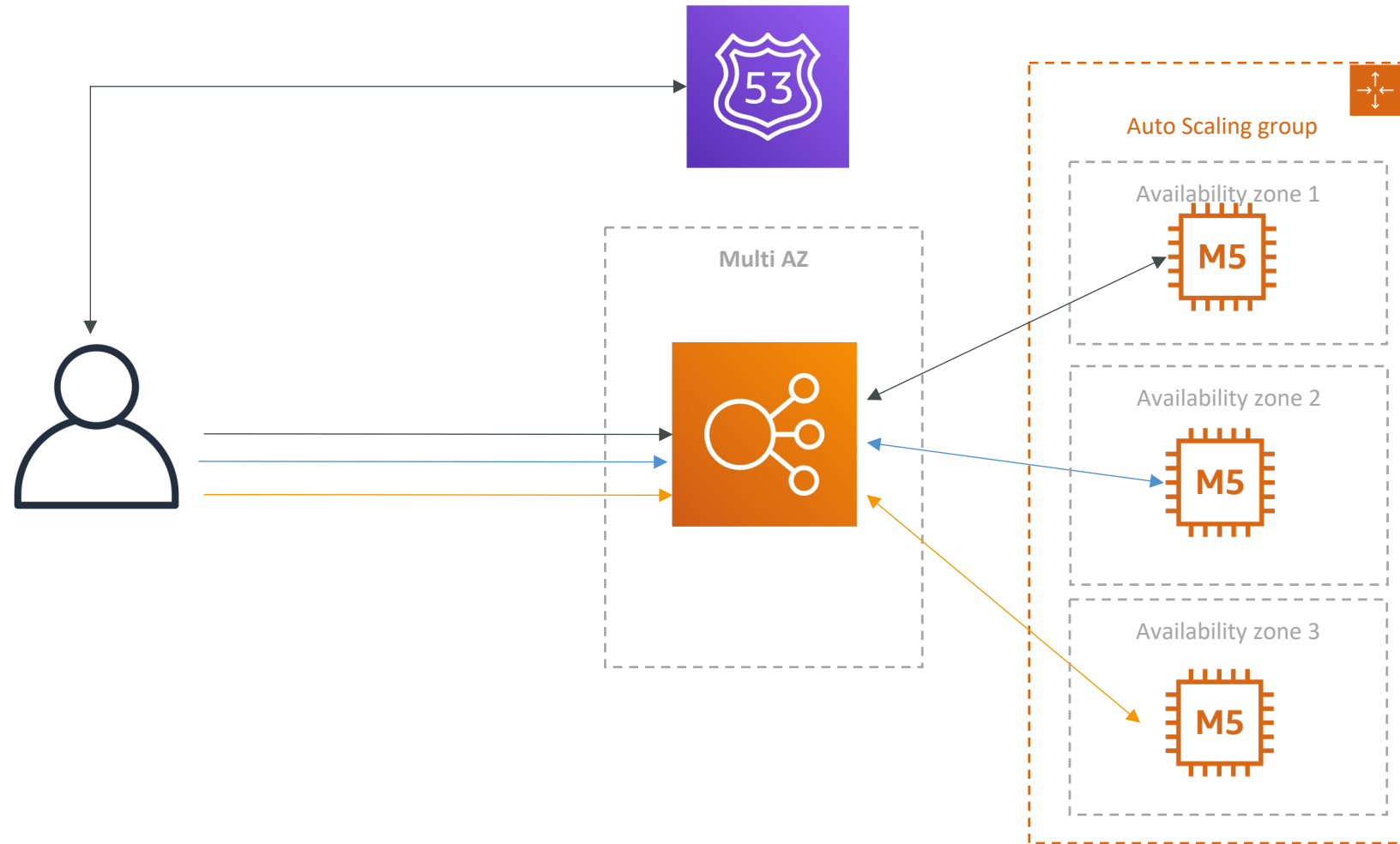
ElastiCache
Multi AZ
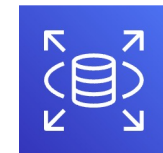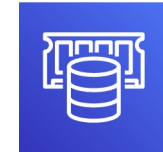
RDS
Multi AZ

Stateful Web App: MyClothes.com Security Groups

Restrict traffic to ElastiCache Security group from the EC2 security group

ElastiCache

Auto Scaling group

Availability zone 1

M5

Multi AZ

Open HTTP / HTTPS to 0.0.0.0/0

Availability zone 2

M5

RDS

Availability zone 3

M5

Restrict traffic to EC2 Security group from the LB

Restrict traffic to RDS Security group from the EC2 security group
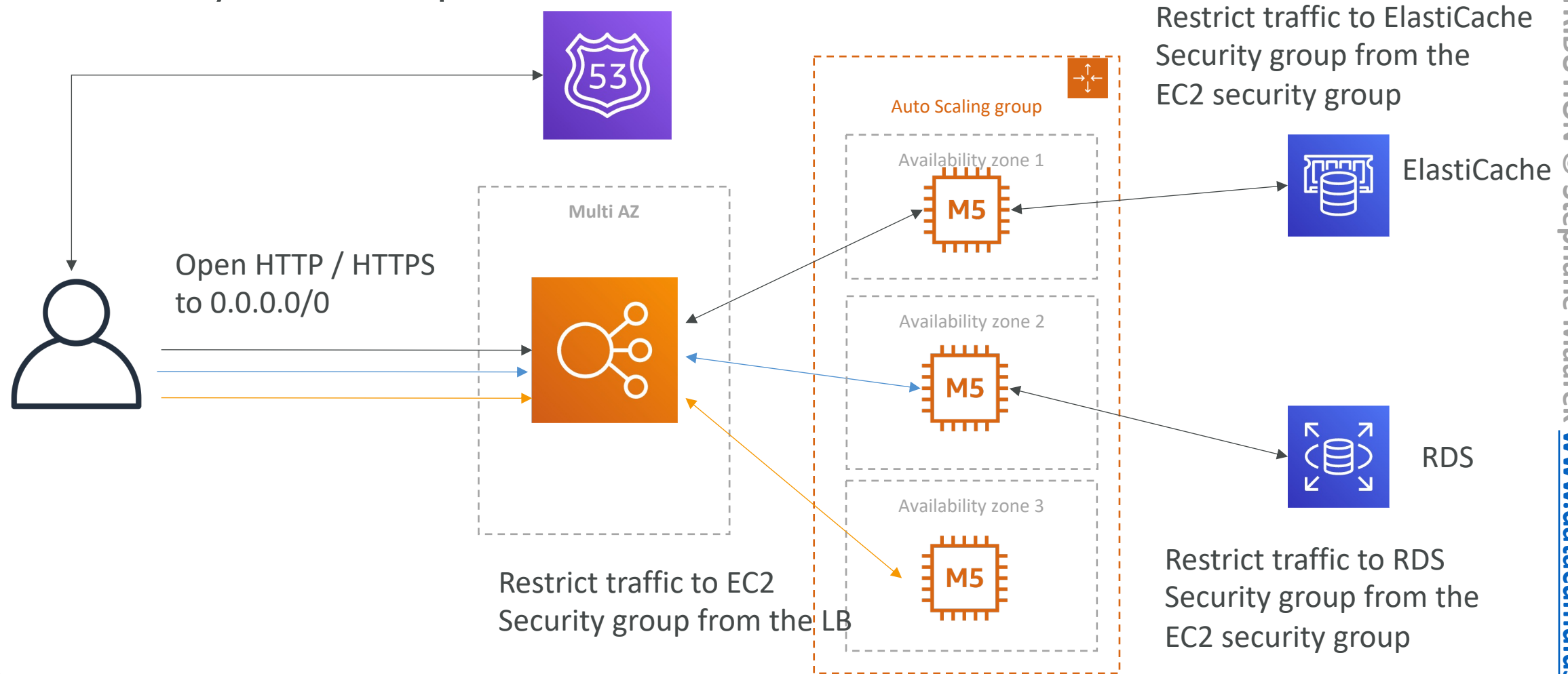
In this lecture we've discussed…
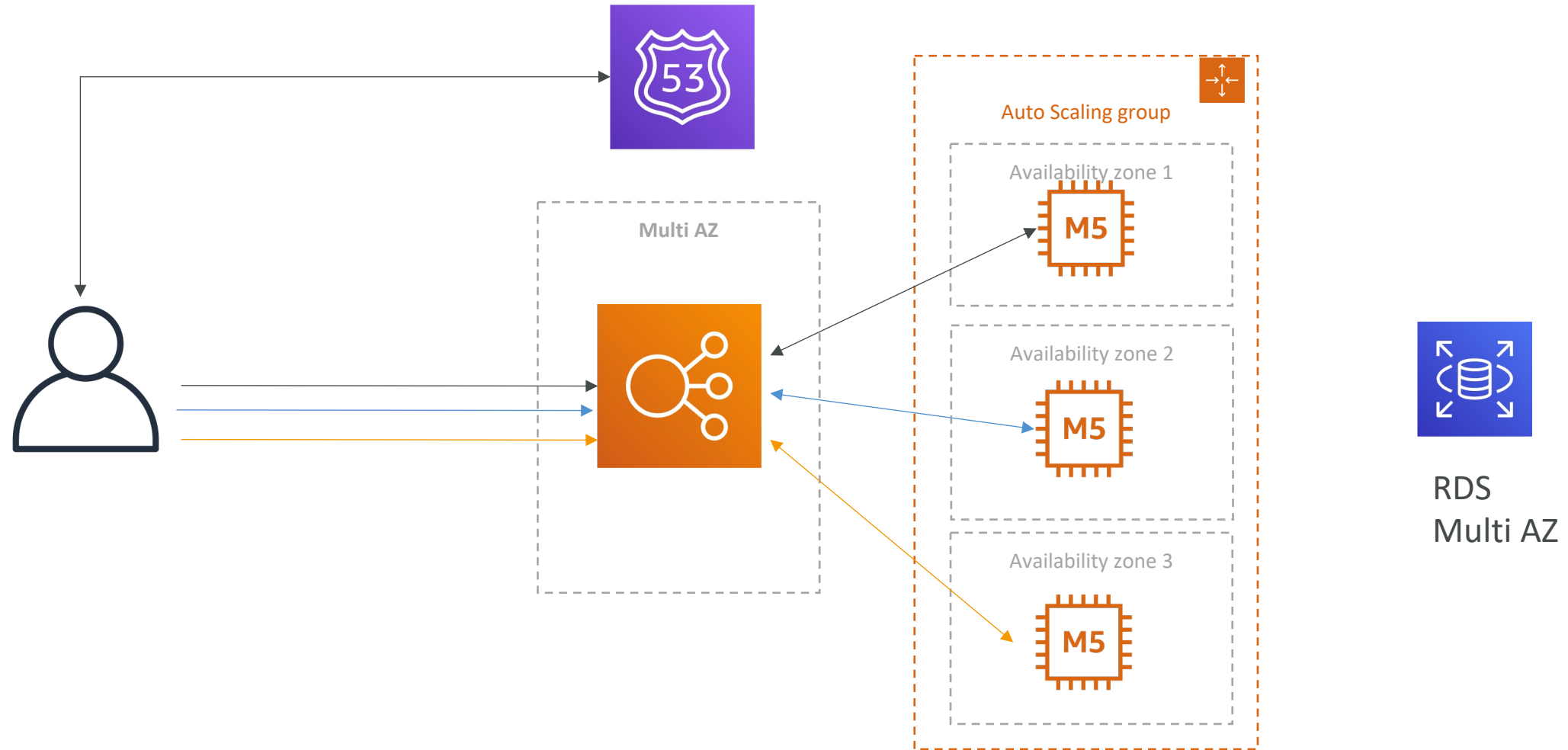# 3-tier architectures for web applications

- ELB sticky sessions

- Web clients for storing cookies and making our web app stateless

- ElastiCache
  - For storing sessions (alternative: DynamoDB)
  - For caching data from RDS
  - Multi AZ

- RDS
  - For storing user data
  - Read replicas for scaling reads
  - Multi AZ for disaster recovery

- Tight Security with security groups referencing each other

Stateful Web App: MyWordPress.com

- We are trying to create a fully scalable WordPress website

- We want that website to access and correctly display picture uploads

- Our user data, and the blog content should be stored in a MySQL database.

- Let's see how we can achieve this!
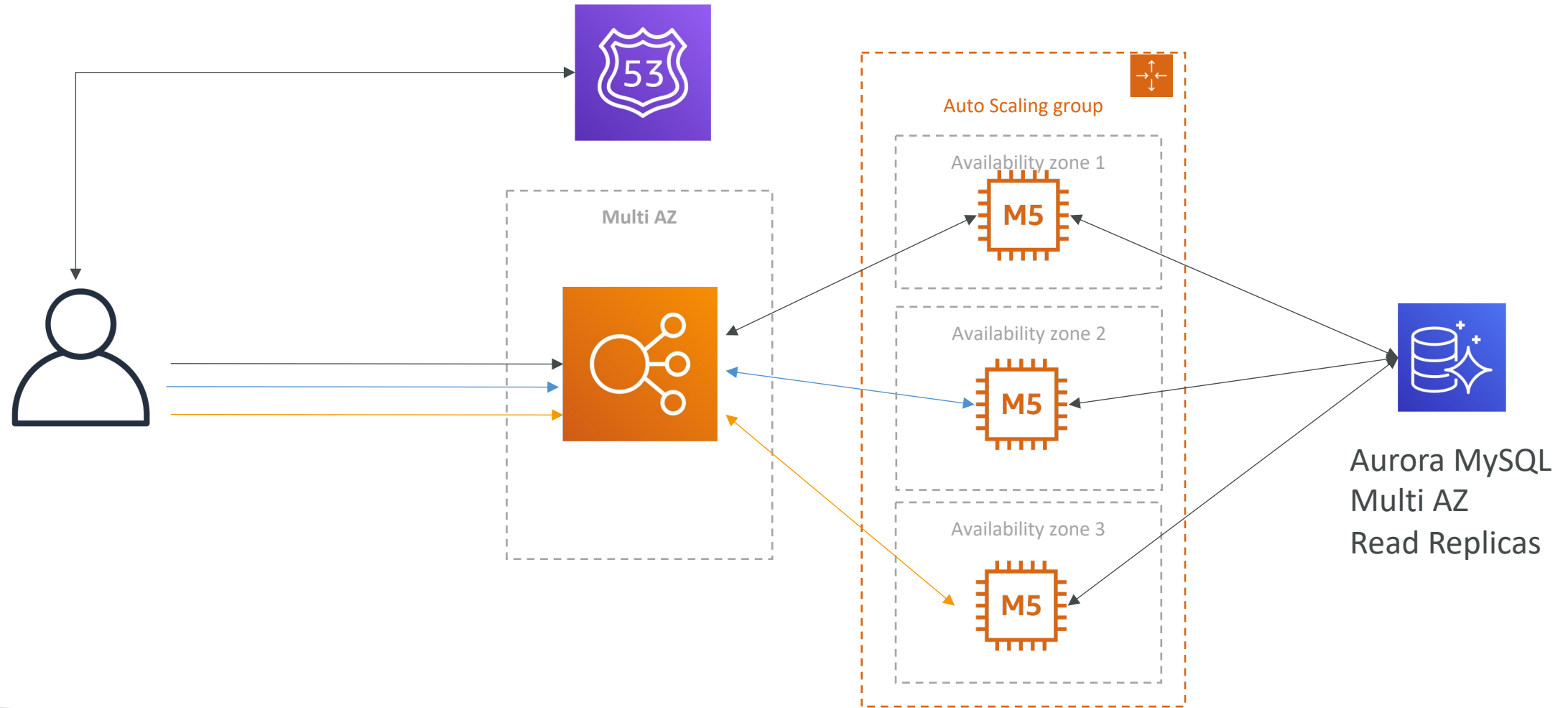
Stateful Web App: MyWordPress.com RDS layer

Stateful Web App: MyWordPress.com
Scaling with Aurora: Multi AZ & Read Replicas

Auto Scaling group

Availability zone 1

Availability zone 2

Availability zone 3

Multi AZ

Aurora MySQL
Multi AZ
Read Replicas

Stateful Web App: MyWordPress.com Storing images with EBS
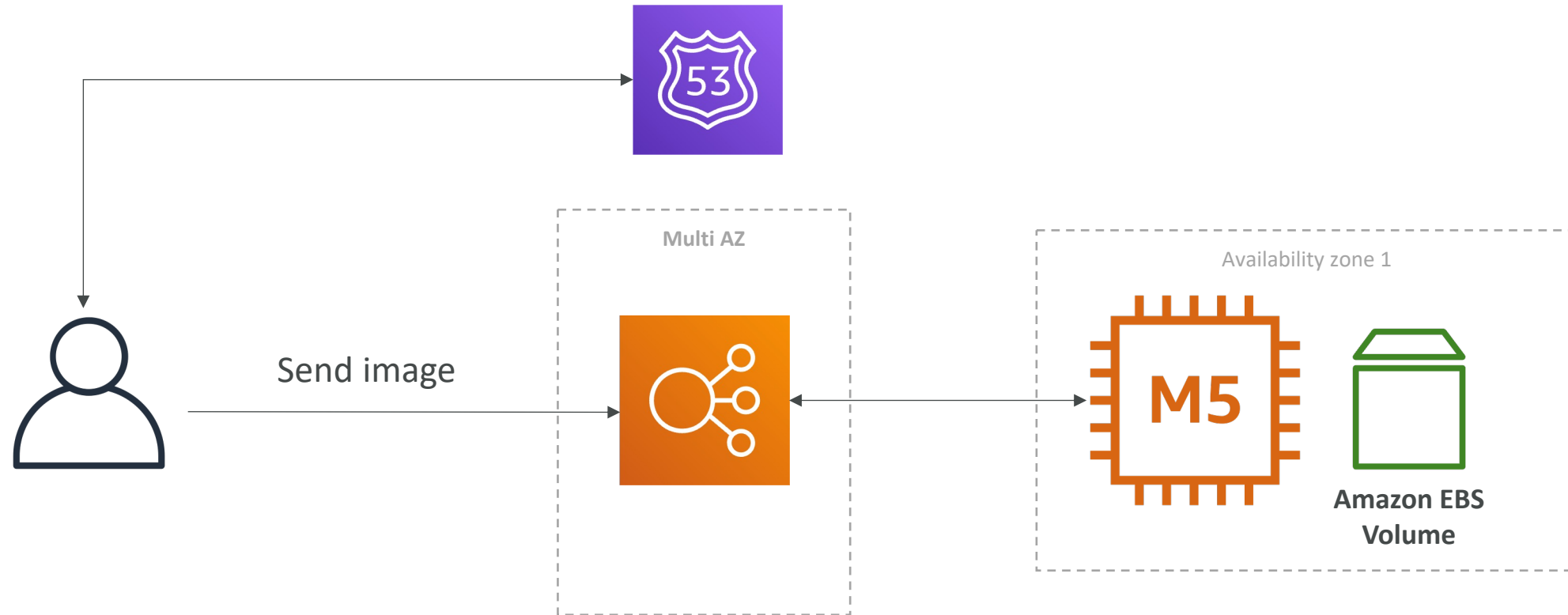
Send image

Multi AZ

Availability zone 1

M5

Amazon EBS Volume

Stateful Web App: MyWordPress.com Storing images with EBS

Send image

Multi AZ

Availability zone 1

M5

Amazon EBS Volume

Availability zone 2

M5

Amazon EBS Volume

Stateful Web App: MyWordPress.com Storing images with EFS

Send image

Multi AZ

Availability zone 1

M5

ENI

Availability zone 2

M5

ENI

EFS

© Stephane Maarek
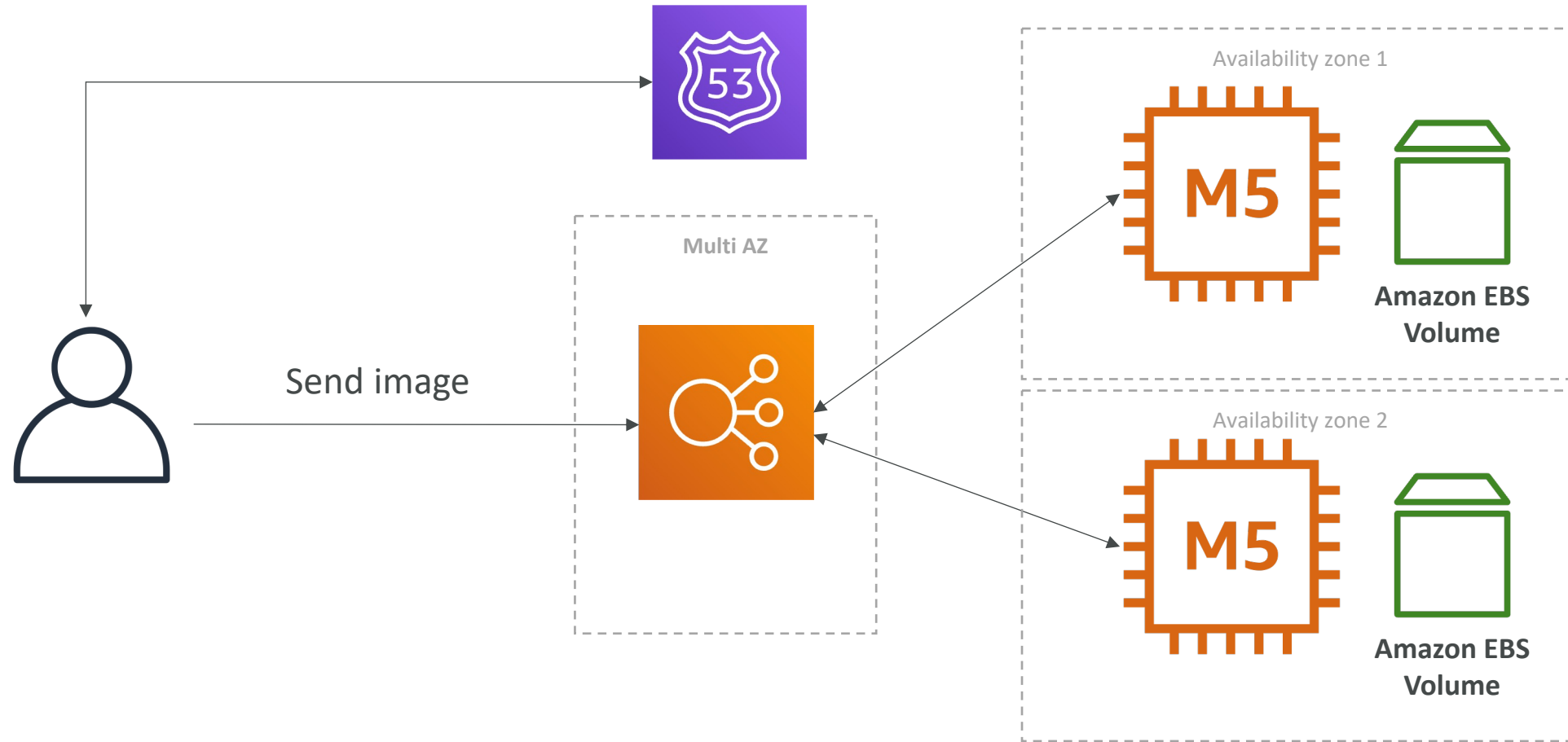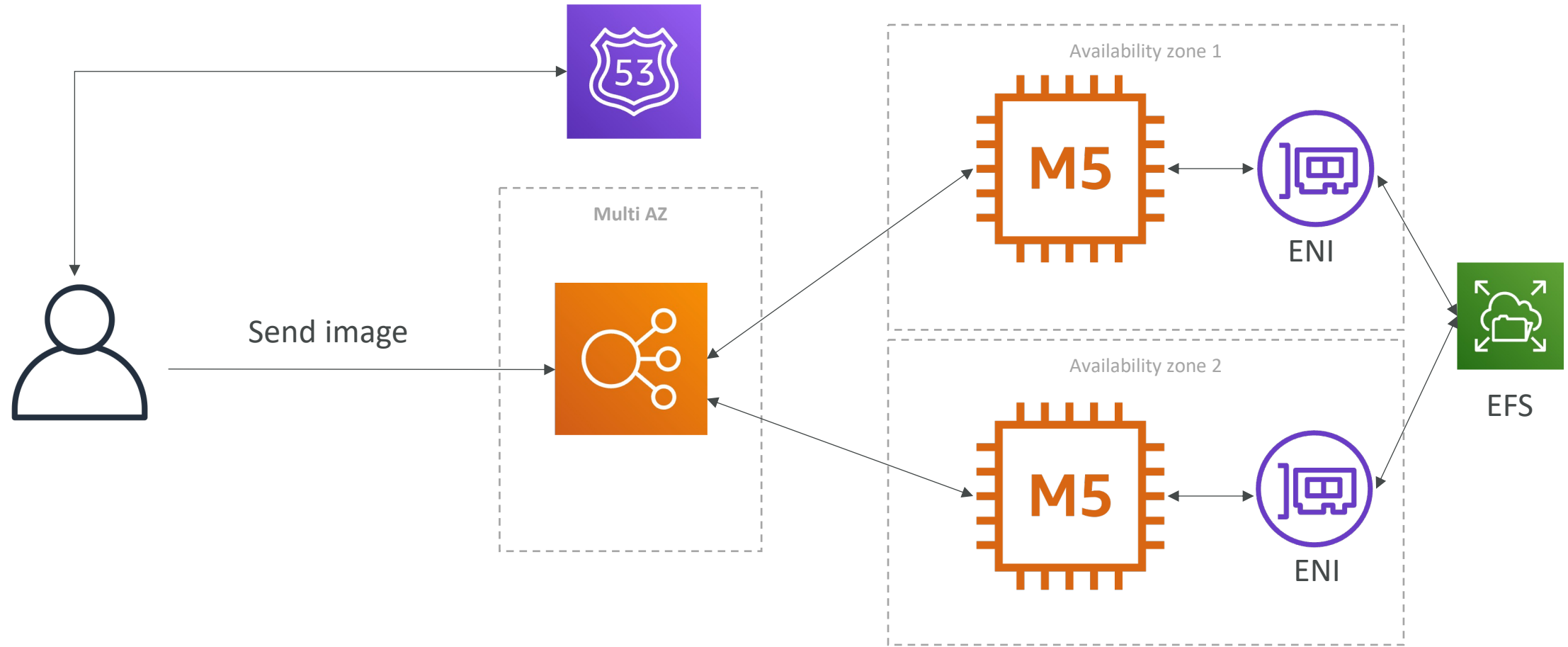
In this lecture we've discussed…

- Aurora Database to have easy Multi-AZ and Read-Replicas
- Storing data in EBS (single instance application)
- Vs Storing data in EFS (distributed application)

Instantiating Applications quickly

- When launching a full stack (EC2, EBS, RDS), it can take time to:
   - Install applications
    - Insert initial (or recovery) data
    - Configure everything
    - Launch the application

- We can take advantage of the cloud to speed that up!

Instantiating Applications quickly

- EC2 Instances:
  - **Use a Golden AMI:** Install your applications, OS dependencies etc.. beforehand and launch your EC2 instance from the Golden AMI
  - **Bootstrap using User Data:** For dynamic configuration, use User Data scripts
  - **Hybrid:** mix Golden AMI and User Data (Elastic Beanstalk)

- RDS Databases:
  - Restore from a snapshot: the database will have schemas and data ready!

- EBS Volumes:
  - Restore from a snapshot: the disk will already be formatted and have data!
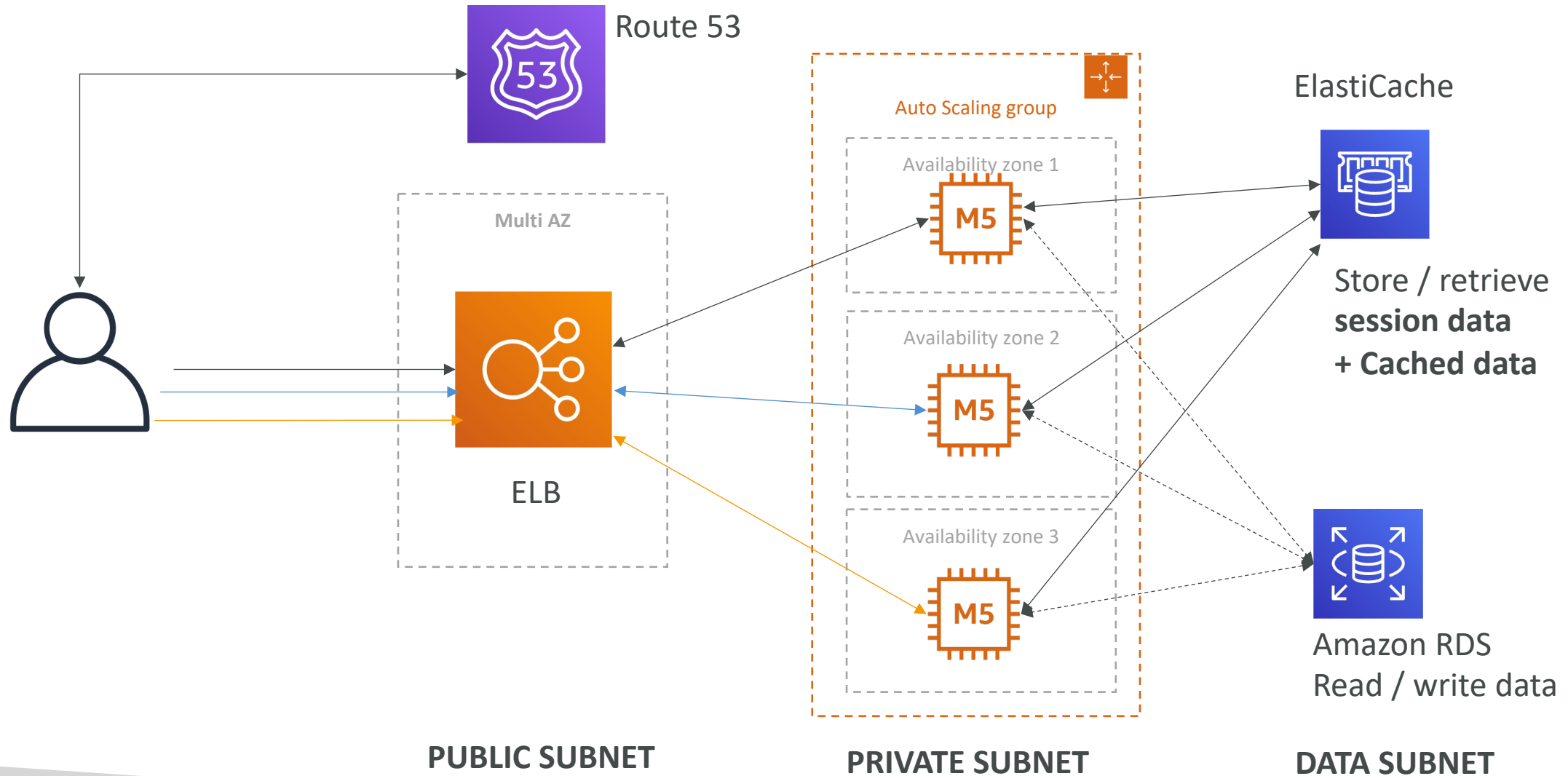
Developer problems on AWS

- Managing infrastructure

- Deploying Code

- Configuring all the databases, load balancers, etc

- Scaling concerns

- Most web apps have the same architecture (ALB + ASG)

- All the developers want is for their code to run!

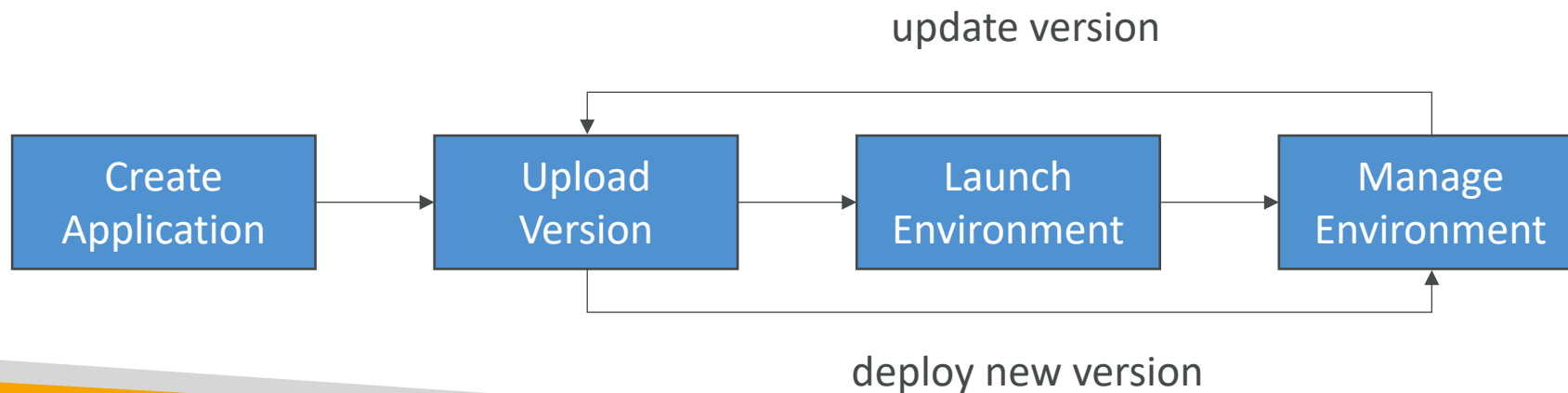- Possibly, consistently across different applications and environments

Elastic Beanstalk – Overview

- Elastic Beanstalk is a developer centric view of deploying an application on AWS

- It uses all the component's we've seen before: EC2, ASG, ELB, RDS, …

- Managed service
  - Automatically handles capacity provisioning, load balancing, scaling, application health monitoring, instance configuration, …
  - Just the application code is the responsibility of the developer

- We still have full control over the configuration

- Beanstalk is free but you pay for the underlying instances
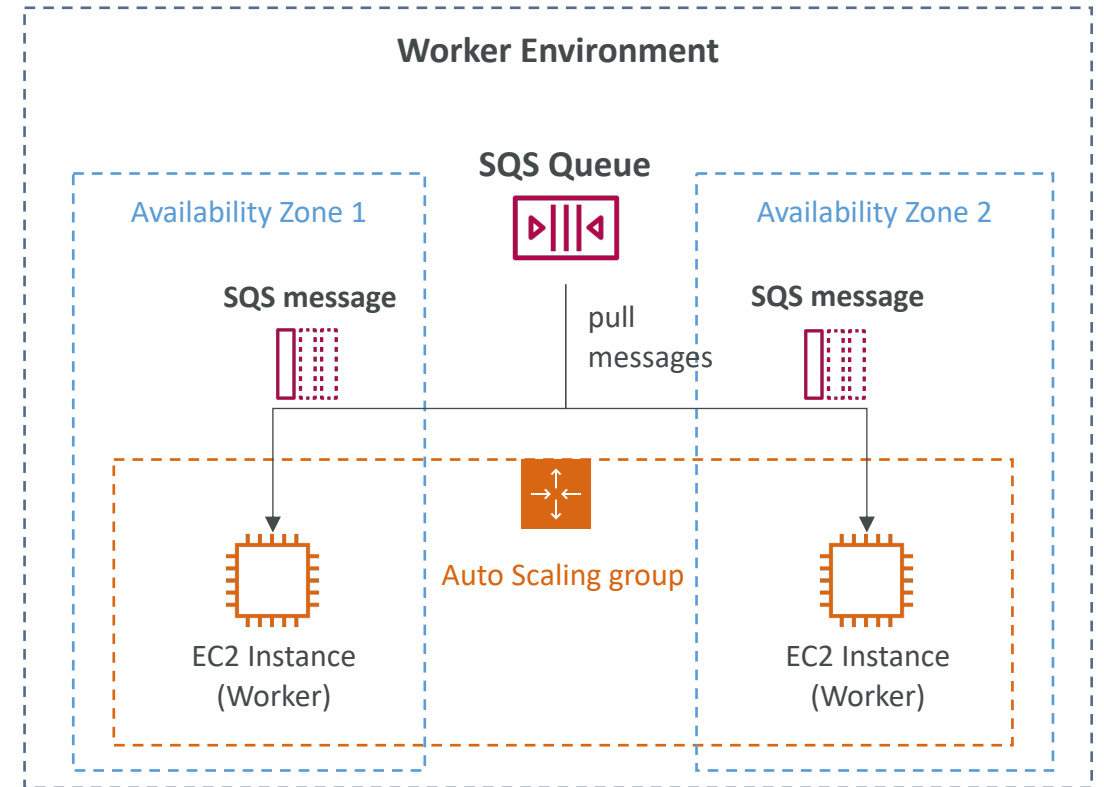
Elastic Beanstalk – Components

- **Application:** collection of Elastic Beanstalk components (environments, versions, configurations, …)
- **Application Version:** an iteration of your application code
- Environment
  - Collection of AWS resources running an application version (only one application version at a time)
  - **Tiers:** Web Server Environment Tier & Worker Environment Tier
  - You can create multiple environments (dev, test, prod, …)

update version

| Create Application | → | Upload Version | → | Launch Environment | → | Manage Environment |

deploy new version

Elastic Beanstalk – Supported Platforms

- Go
- Java SE
- Java with Tomcat
- .NET Core on Linux
- .NET on Windows Server
- Node.js
- PHP
- Python

- Ruby
- Packer Builder
- Single Container Docker
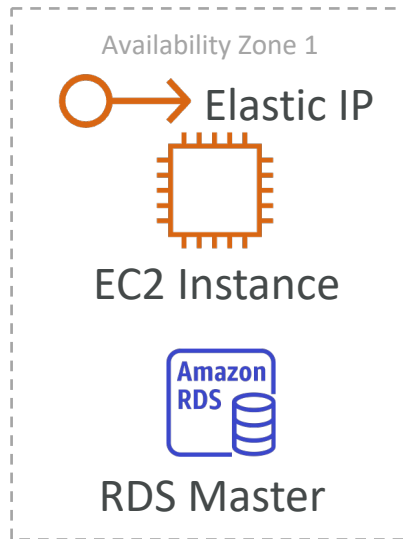- Multi-container Docker
- Preconfigured Docker

Web Server Tier vs. Worker Tier

Web Environment
(myapp.us-east-1.elasticbeanstalk.com)

Availability Zone 1

ELB

Availability Zone 2

Security Group

EC2 Instance
(Web Server)

Auto Scaling group

Security Group

EC2 Instance
(Web Server)

Worker Environment

SQS Queue

Availability Zone 1

SQS message

pull messages

Availability Zone 2

SQS message

EC2 Instance
(Worker)

Auto Scaling group

EC2 Instance
(Worker)

- Scale based on the number of SQS messages
- Can push messages to SQS queue from another Web Server Tier

Elastic Beanstalk Deployment Modes

Single Instance
Great for dev

High Availability with Load Balancer
Great for prod

Availability Zone 1

Elastic IP

EC2 Instance

Amazon RDS

RDS Master

ALB

Availability Zone 1

Availability Zone 2

Auto Scaling Group

EC2 Instance

EC2 Instance

Amazon RDS

Amazon RDS

RDS Master

RDS Standby