



CSCI 5902 Adv. Cloud Architecting
Fall 2023
Instructor: Lu Yang

Module 12 Disaster recovery (Section 3) &
Module 13 Caching contents (Sections 1-3)
Dec 1, 2023

Housekeeping items and feedback

1. Start recording
2. The last lab assignment due on Dec 1
3. The last architecture assignment due on Dec 4
4. SLEQ



Recap of our last lecture

Module overview



Sections

1. Architectural need
 2. Disaster planning strategies
 3. Disaster recovery patterns
- We stopped here in the last lecture**

Module 14: Planning for Disaster

Section 3: Disaster recovery patterns

Common disaster recovery patterns on AWS



Four disaster recovery patterns

- Backup and restore
- Pilot light
- Warm standby
- Multi-site



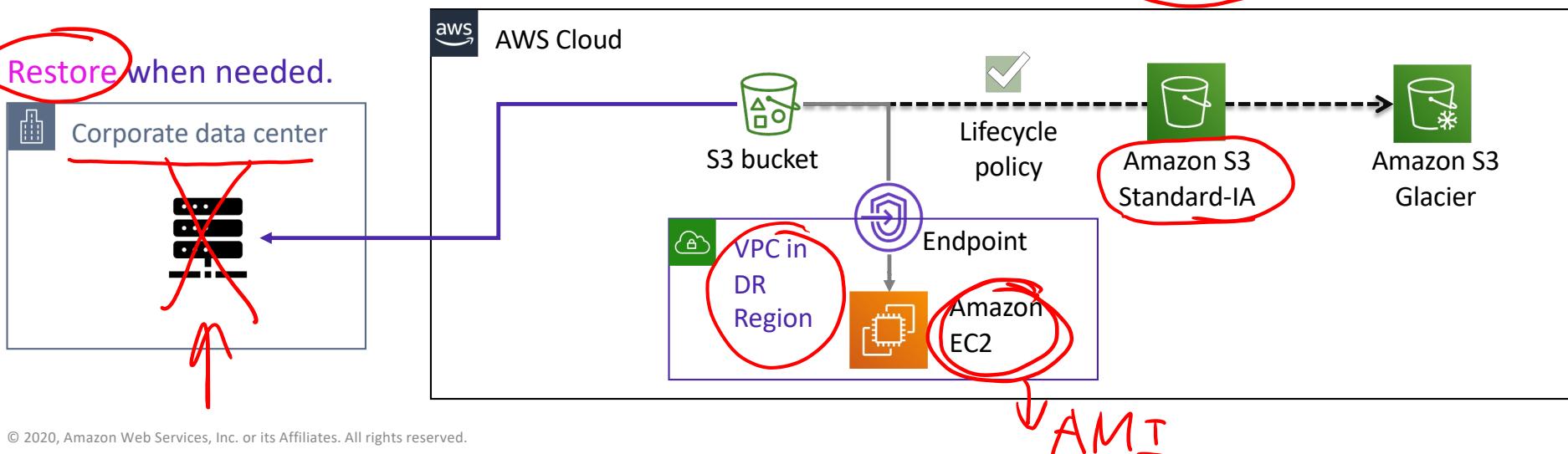
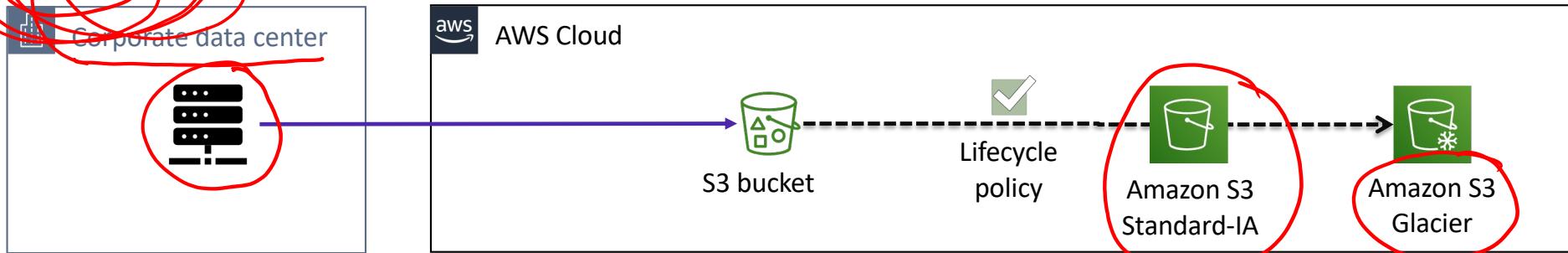
Each pattern is suited to a different combination of:

- Recovery point objective *RPO*
- Recovery time objective *RTO*
- Cost-effectiveness

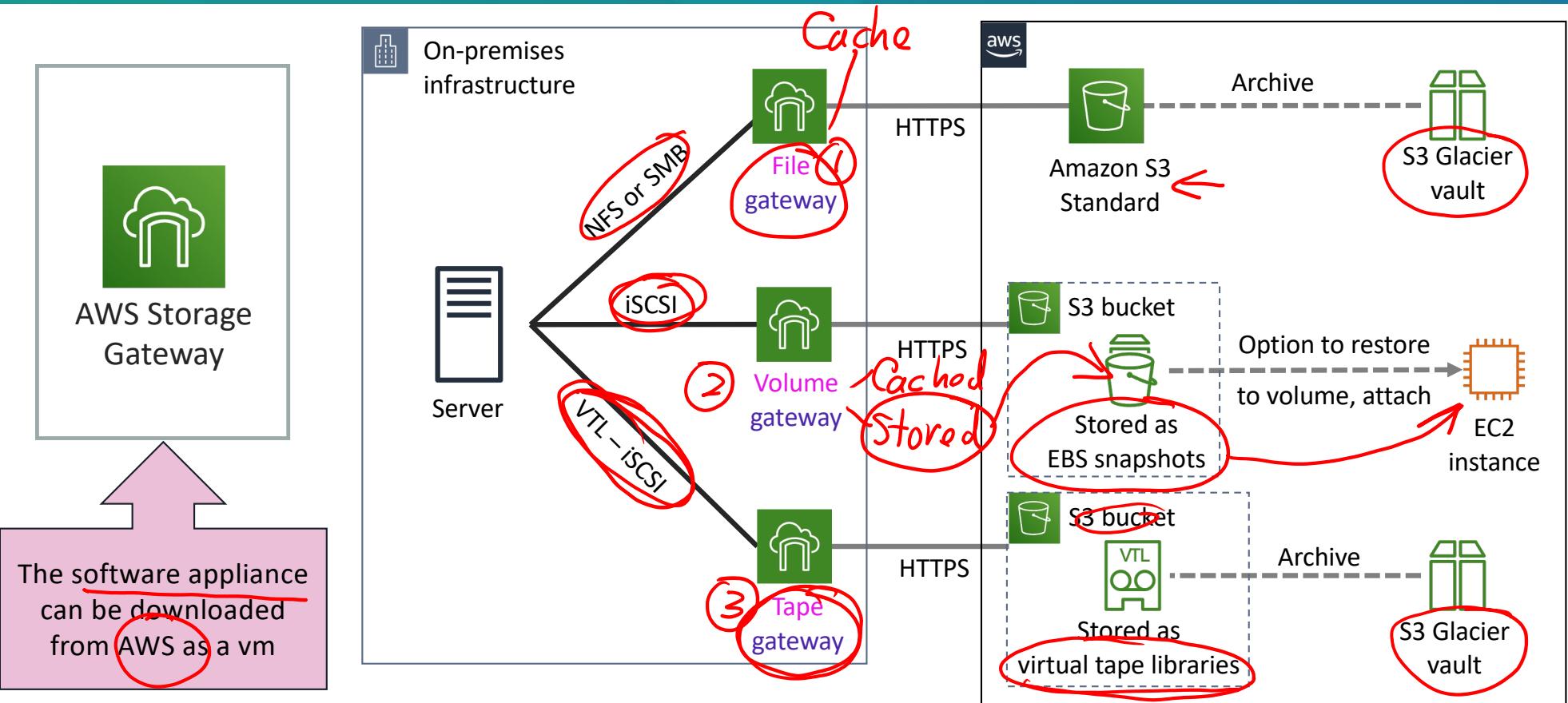
① Backup and restore pattern



Back up configuration and state data to S3. Implement lifecycle policy to save on cost.



AWS Storage Gateway



• Backup and restore: Checklist



Preparation phase

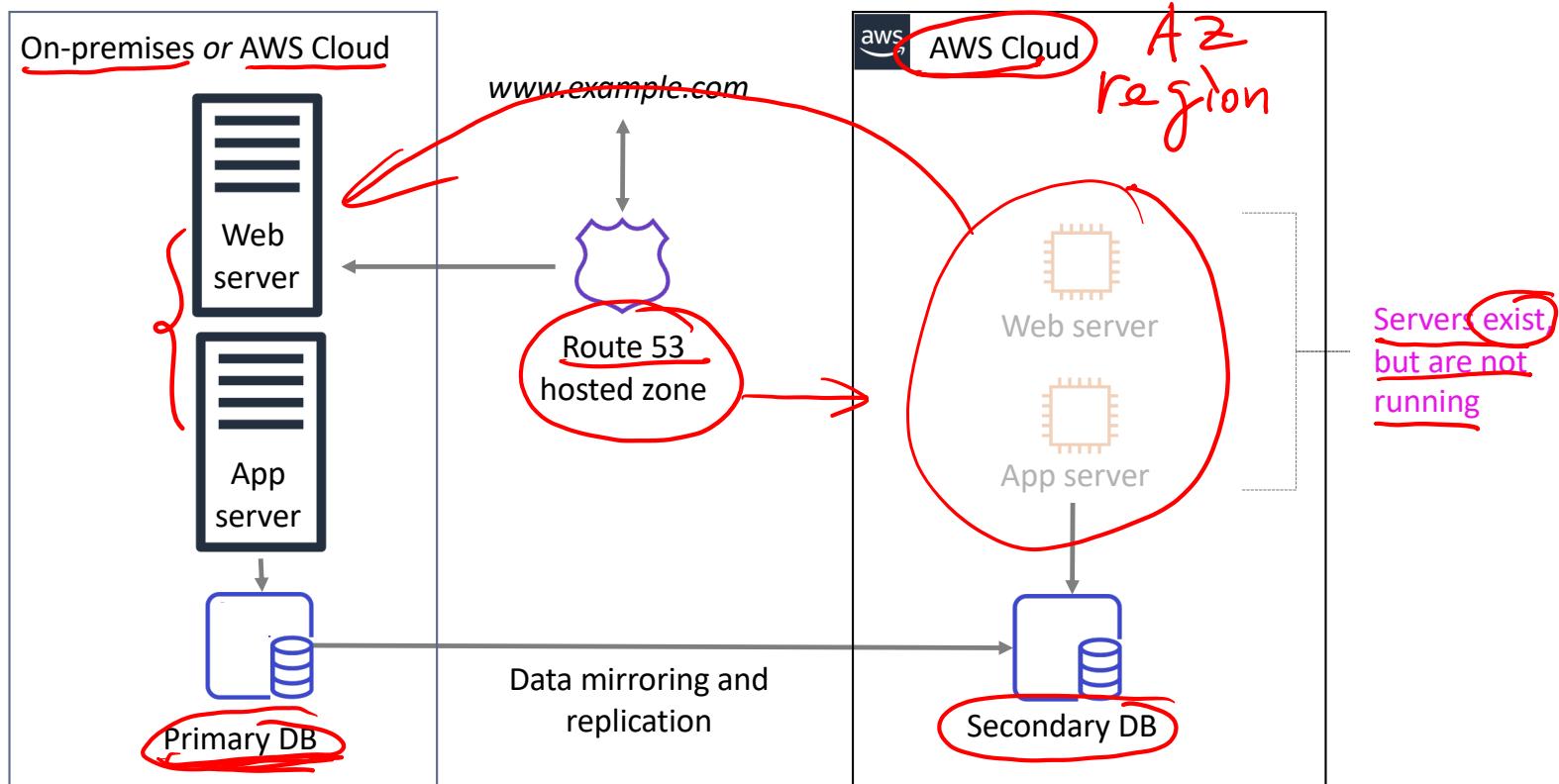
- Create backups of current systems
- Store backups in Amazon S3
- Document procedure to restore from backups
- Know:
 - Which AMI to use, and build as needed
 - How to restore system from backups
 - How to route traffic to the new system
 - How to configure the deployment

Test

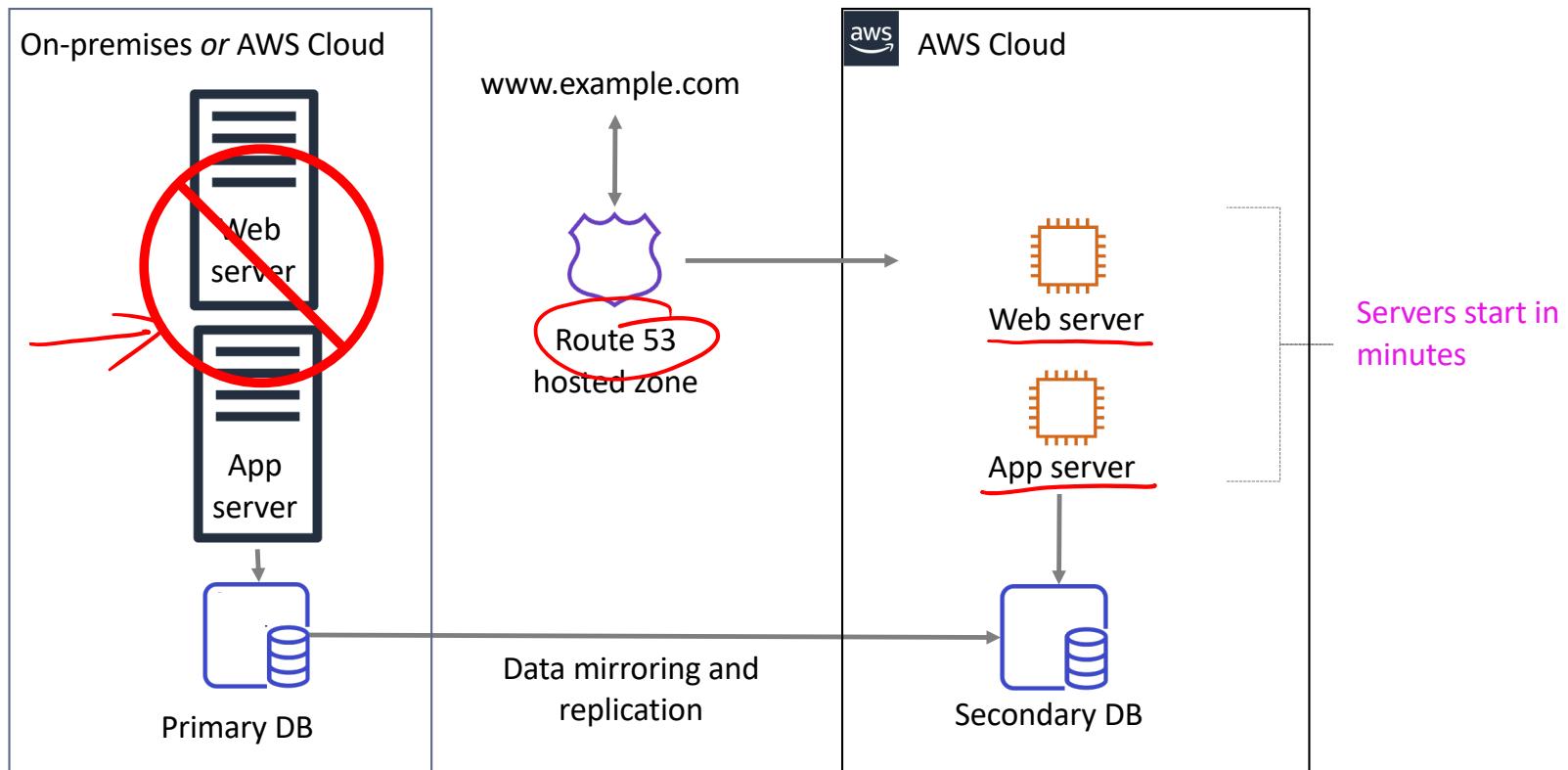
In case of disaster

- Retrieve backups from Amazon S3
- Restore required infrastructure
 - EC2 instances from prepared AMIs
 - Elastic Load Balancing load balancers
 - AWS resources created by an AWS CloudFormation stack – automated deployment to restore or duplicate the environment
↓ CI/CD
- Restore system from backup
- Route traffic to the new system
 - Adjust Domain Name System (DNS) records accordingly

② Pilot light pattern: Preparation phase



Pilot light pattern: In case of disaster



Pilot light pattern: Checklist



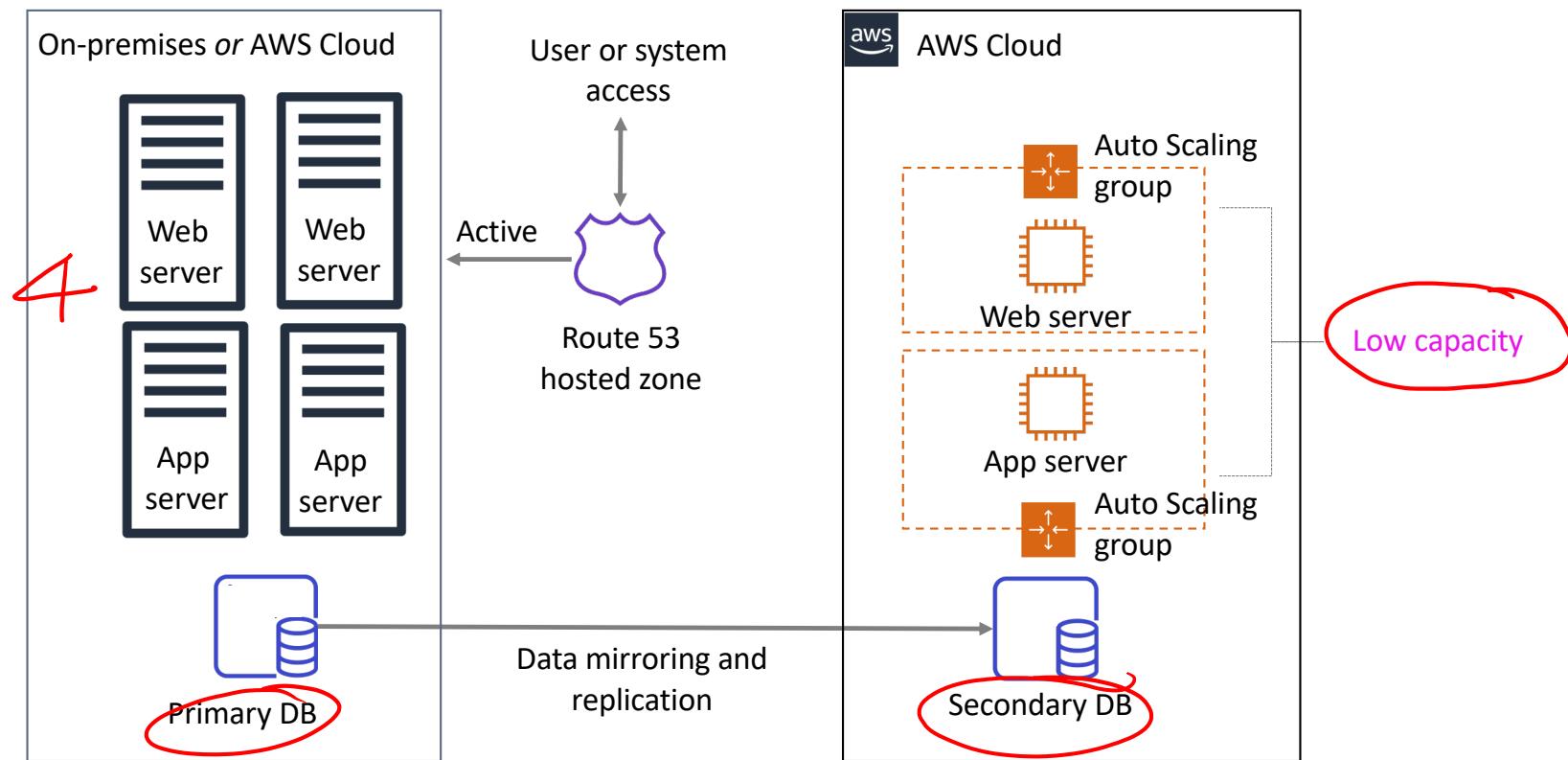
Preparation phase

- Configure EC2 instances to replicate or mirror servers
- Ensure that all supporting custom software packages are available on AWS
- Create and maintain AMIs of key servers where fast recovery is needed
- Regularly run these servers, test them, and apply any software updates and configuration changes
- Consider automating the provisioning of AWS resources

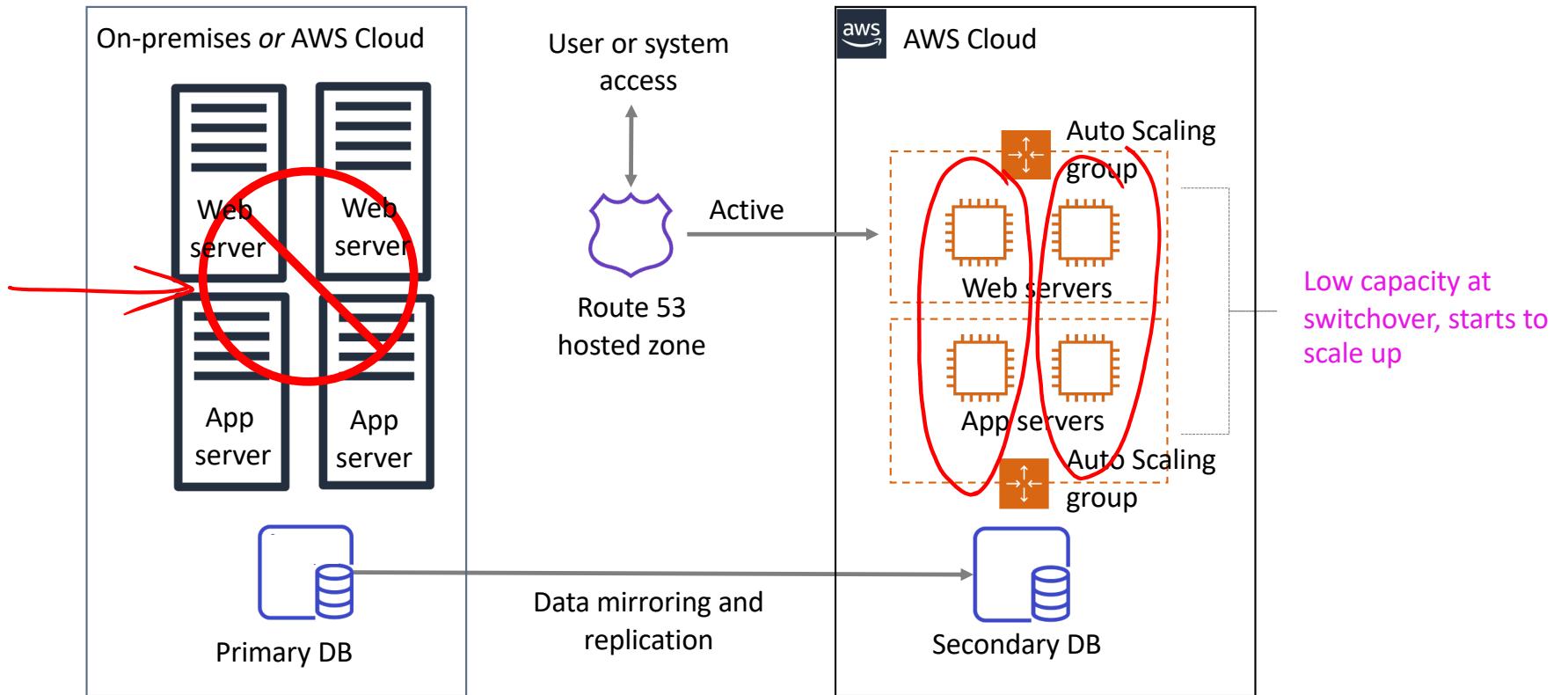
In case of disaster

- Automatically bring up resources around the replicated core dataset
- Scale the system as needed to handle current production traffic
- Switch over to the new system
 - Adjust DNS records to point to AWS

② Warm standby pattern: Preparation phase



Warm standby pattern: In case of disaster



Warm standby pattern: Checklist



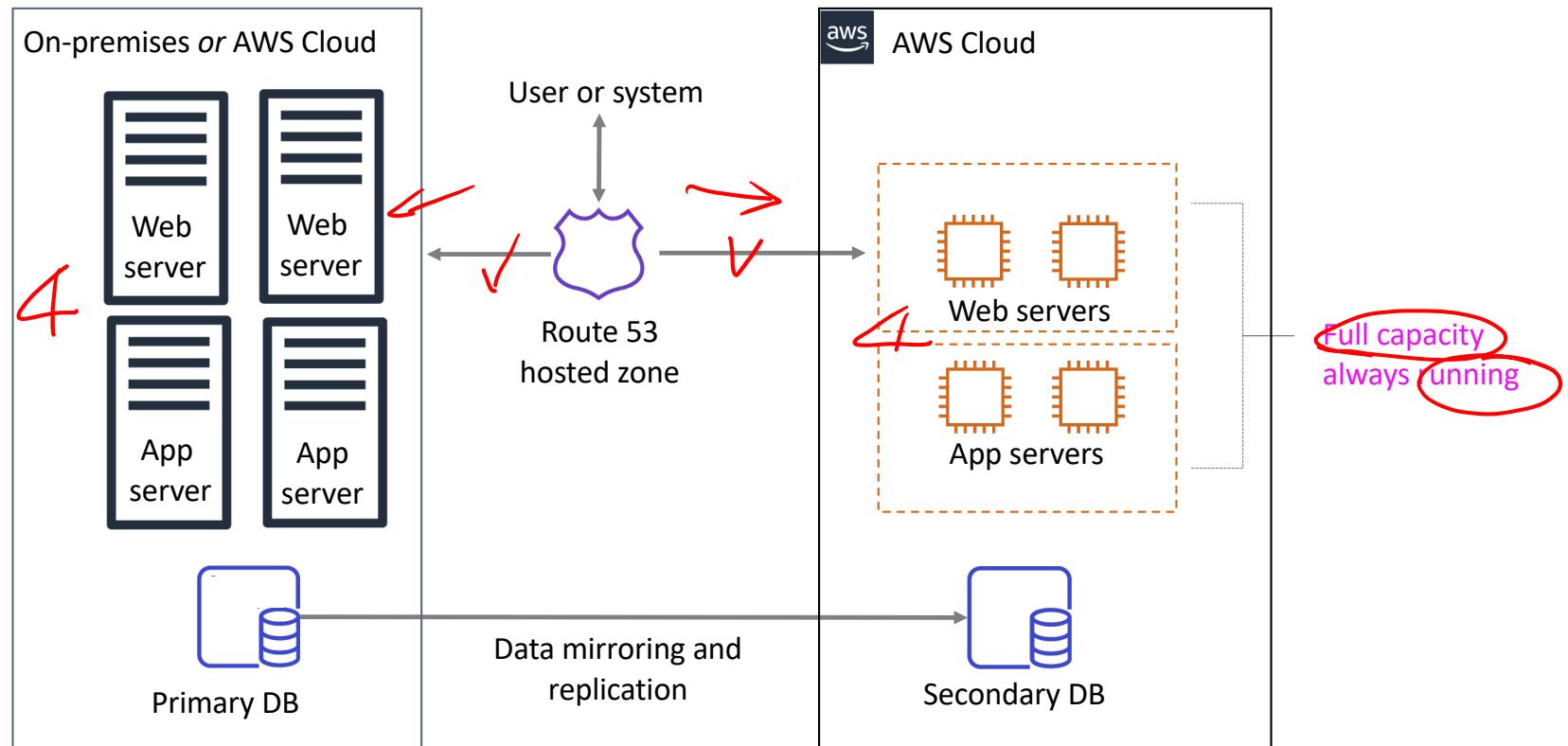
Preparation

- Similar to pilot light
- All necessary components running 24/7, but not scaled for production traffic
- Best practice: Continuous testing
 - Trickle a statistical subset of production traffic to the DR site

In case of disaster

- Immediately fail over most critical production load
 - Adjust DNS records to point to AWS
- (Automatically) Scale the system further to handle all production load

④ Multi-site pattern



Multi-site: Checklist



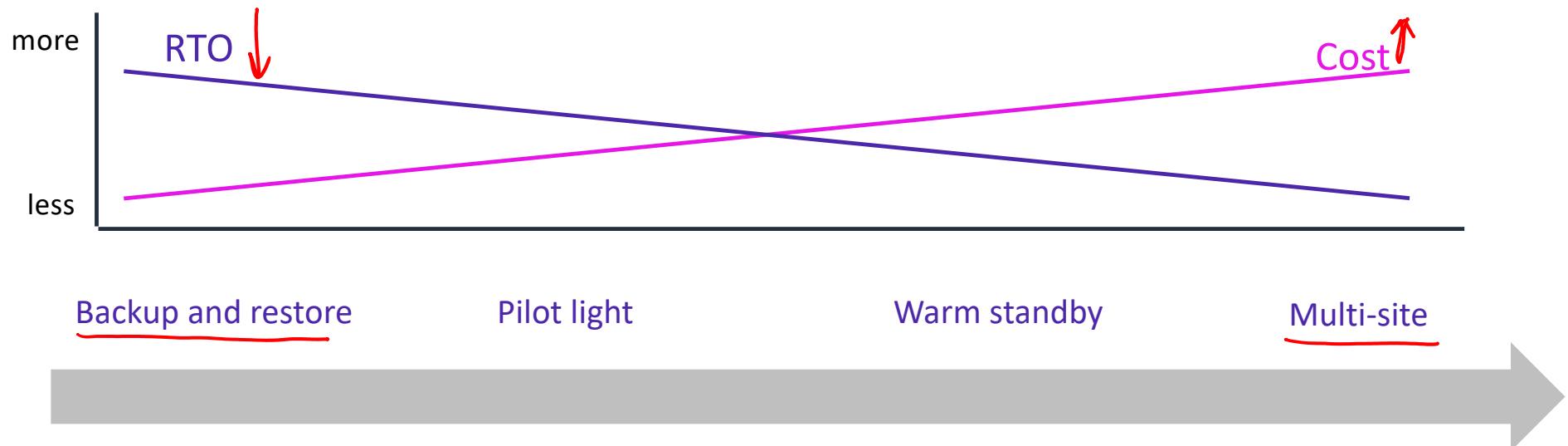
Preparation

- Similar to warm standby
- Configured for full scaling in or scaling out for production load

In case of disaster

- Immediately fail over all production load

Summary of common DR patterns



- Lower-priority use cases
- Solutions: Amazon S3, Storage Gateway

- Meeting lower RTO and RPO requirements
- Core services
- Scale AWS resources in response to a DR event

- Solutions that require RTO and RPO in minutes
- Business-critical services

- Automatic failover of your environment in AWS to a running duplicate

DR preparation: Best practices



Start simple



Check for software licensing issues



Practice Game Day exercises

Section 3 key takeaways



20



- Common [disaster recovery patterns](#) on AWS include backup and restore, pilot light, warm standby, and multi-site.
- [Backup and restore](#) is the most cost effective approach. However, it has the highest RTO.
- [Multi-site](#) provides the fastest RTO. However, it costs the most because it provides a fully running production-ready duplicate.
- [AWS Storage Gateway](#) provides three interfaces—file gateway, volume gateway, and tape gateway—for data backup and recovery between on-premises and the AWS Cloud.

Module 12: Planning for Disaster

Module wrap-up

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Module summary



In summary, in this module, you learned how to:

- Identify strategies for disaster planning
- Define RPO and RTO
- Describe four common patterns for backup and disaster recovery and how to implement them
- Use AWS Storage Gateway for on-premises-to-cloud backup solutions

AWS Academy Cloud Architecting

Module 13: Caching Content

Module overview



Sections

- 1. Architectural need
- 2. Overview of caching
- 3. Edge caching
- 4. Caching web sessions
- 5. Caching databases

Lab

- Guided Lab: Streaming Dynamic Content Using Amazon CloudFront

Module objectives



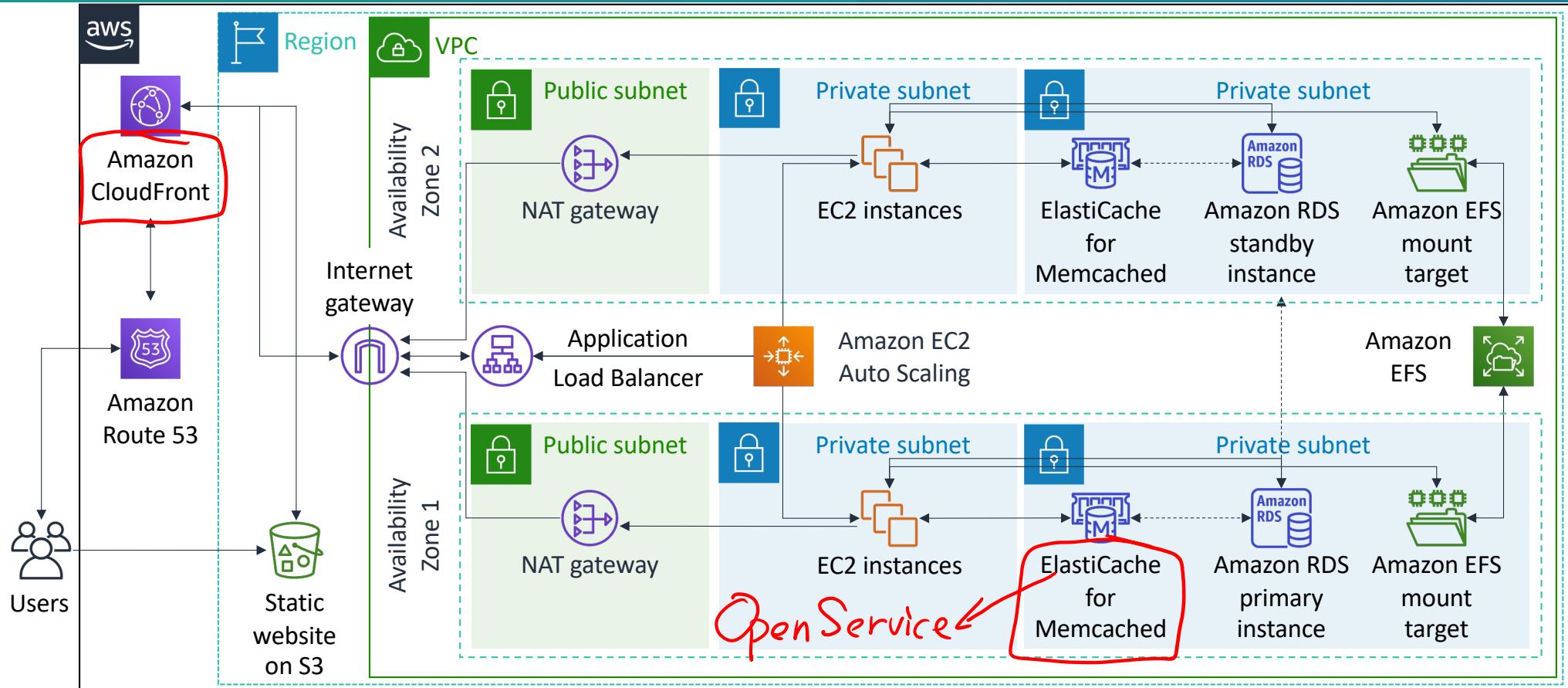
At the end of this module, you should be able to:

- Identify how caching content can improve application performance and reduce latency
- Identify how to design architectures that use edge locations for distribution and distributed denial of service (DDoS) protection
- Create architectures that use Amazon CloudFront to cache content
- Recognize how session management relates to caching
- Describe how to design architectures that use Amazon ElastiCache

Module 11: Caching Content

Section 1: Architectural need

Caching as part of a larger architecture



Café business requirement



The capacity of the café's infrastructure is constantly being overloaded with the same requests. This inefficiency is increasing cost and latency.



Module 11: Caching Content

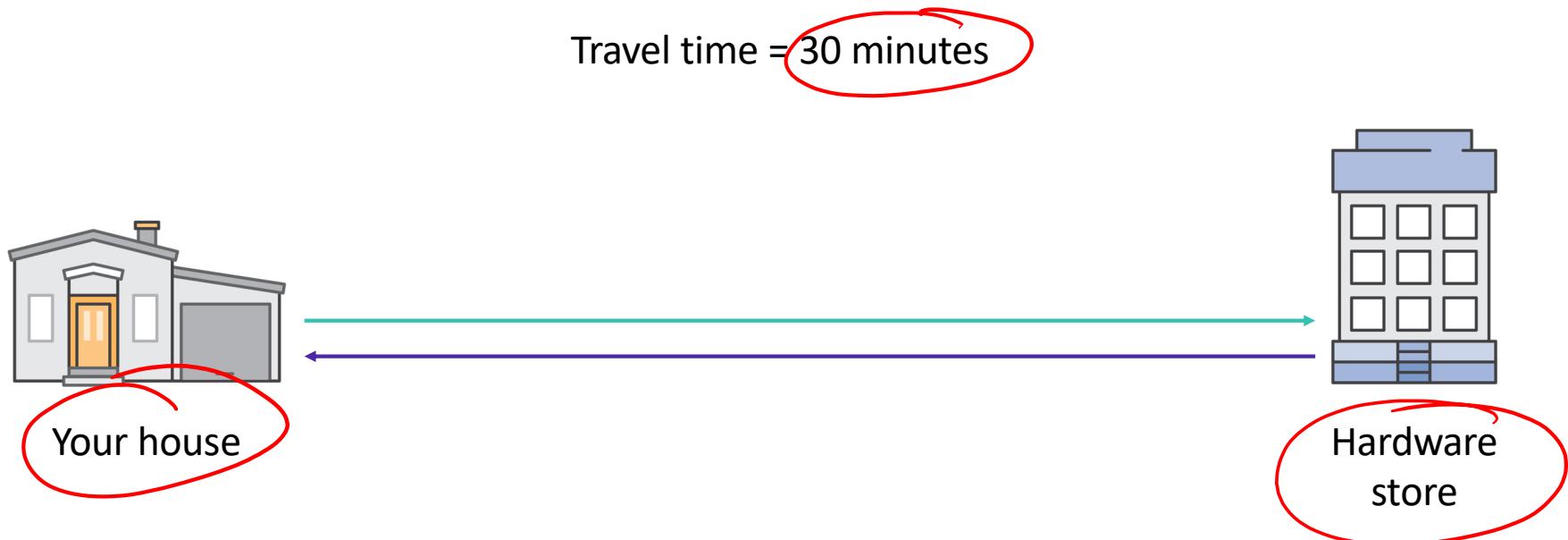
Section 2: Overview of caching

Caching: Trading capacity for speed

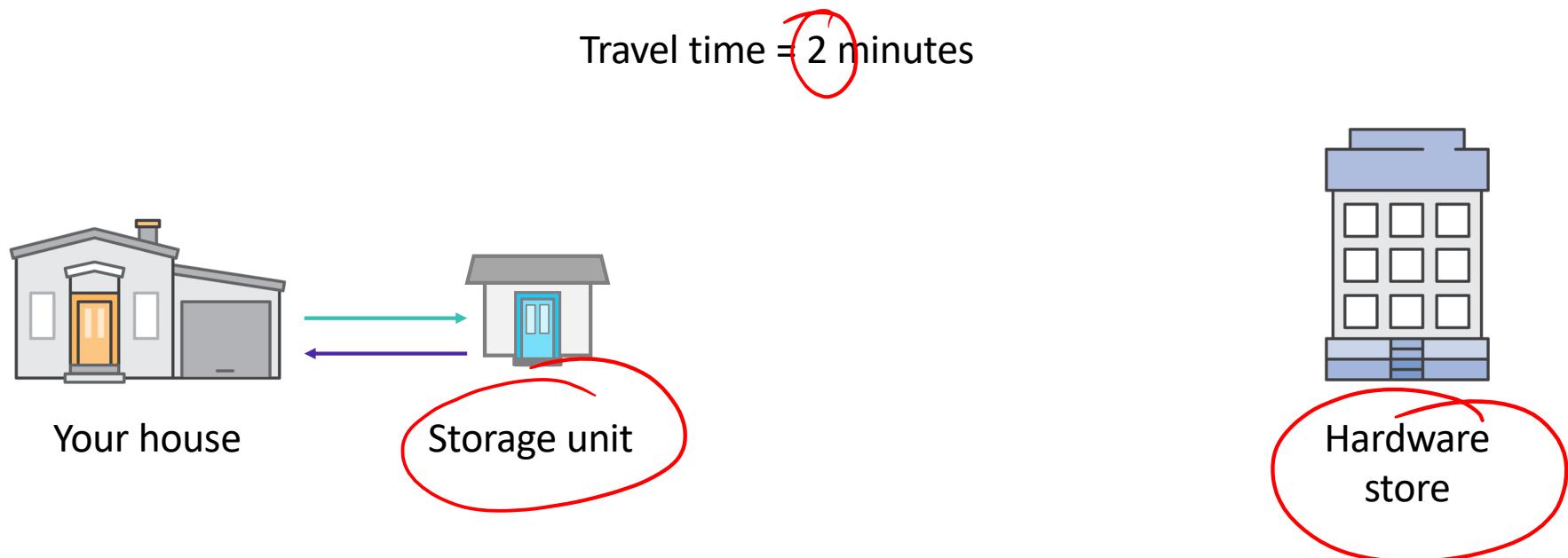


- Is a high-speed data storage layer
- Stores a subset of data
- Increases data retrieval performance
- Reduces the need to access the underlying slower storage layer

Cache example (1 of 2)



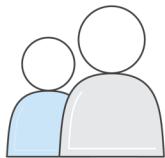
Cache example (2 of 2)



What should you cache?



Data that requires a slow and expensive query to acquire



Relatively static and frequently accessed data—for example, a profile for your social media website

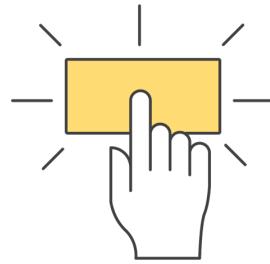


Information that can be stale for some time, such as a publicly traded stock price

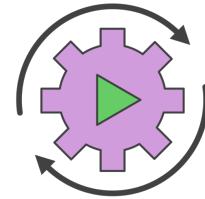
Benefits of caching



Improves application speed



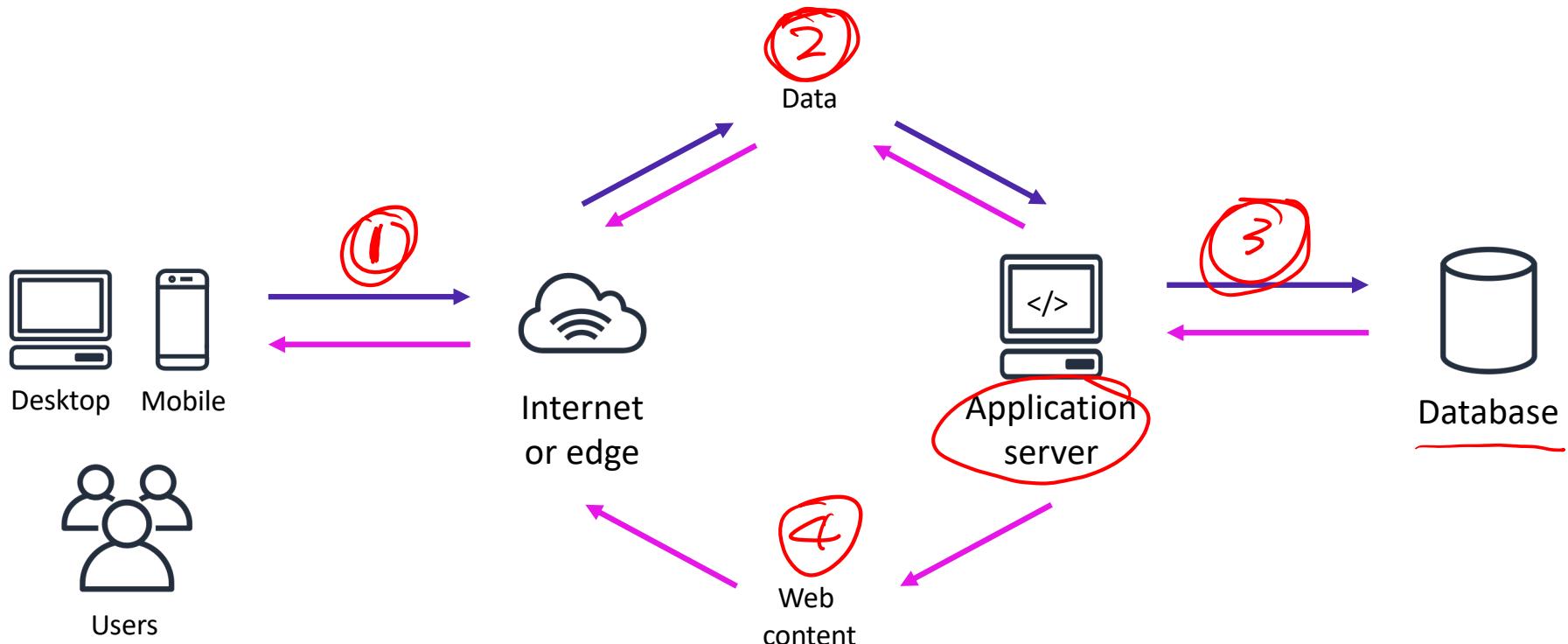
Reduces response latency



Reduces database access time

Speed

Caching throughout the data journey



Section 2 key takeaways



36

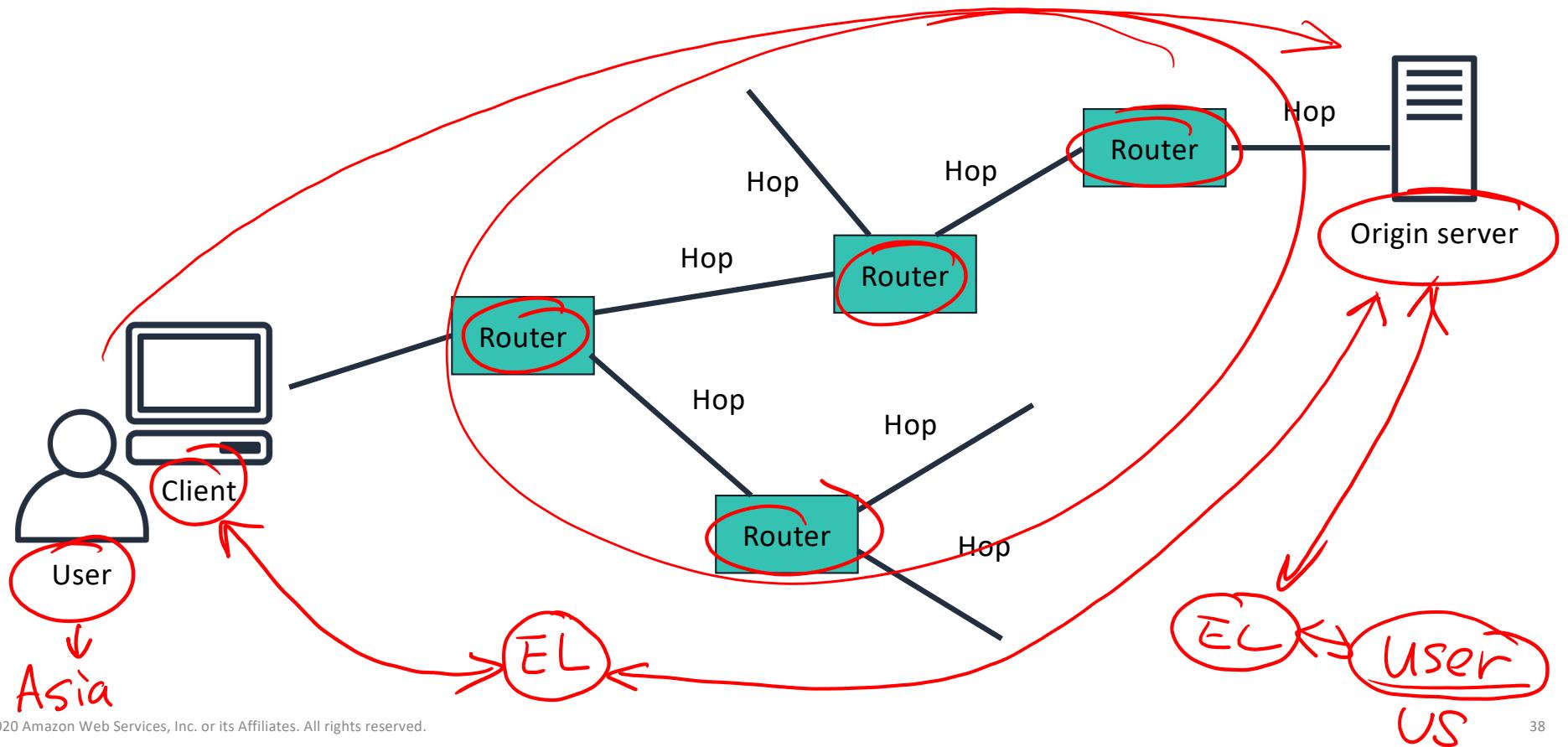


- A cache provides high throughput, low-latency access to commonly accessed application data by storing the data in memory
- When you decide what data to cache, consider speed and expense, data and access patterns, and your application's tolerance for stale data
- Caches can be applied and used throughout various layers of technology, including operating systems, networking layers, web applications, and databases

Module 11: Caching Content

Section 3: Edge caching

Network latency



Content delivery network (CDN)



- Is a globally distributed system of caching servers
- Caches copies of commonly requested files (static content)
- Delivers a local copy of the requested content from a nearby cache edge or Point of Presence
- Improves application performance and scaling



Amazon
CloudFront

- Is the Amazon global CDN
- Is optimized for all delivery use cases, with a multi-tier cache by default and extensive flexibility
- Provides an extra layer of security for your architectures
- Supports WebSockets and HTTP or HTTPS methods

What type of content can you cache in an edge cache?



Secure

Dynamic

Image

Can be
cached!

The diagram illustrates various types of content that can be cached in an edge cache. It shows a screenshot of an Amazon Kindle Fire HDX tablet displaying a live support session with an expert. The screen is divided into several sections:

- Secure:** Points to the HTTPS:// URL in the browser address bar.
- User input:** Points to the search bar and the "Go" button.
- Web objects:** Points to the search results and other web content.
- Can be cached!**: A green callout box points to the image on the screen.
- Dynamic:** Points to the search bar and the "Go" button.
- Image:** Points to the image on the screen.
- Can be cached!**: A green callout box points to the image on the screen.
- Video:** Points to the video player interface on the right.
- Can be cached!**: A green callout box points to the video player interface.

Revolutionary on-device tech support

Exclusively on Kindle Fire HDX tablets—live on-device tech support from an Amazon expert is just a tap away with the new "Mayday" button

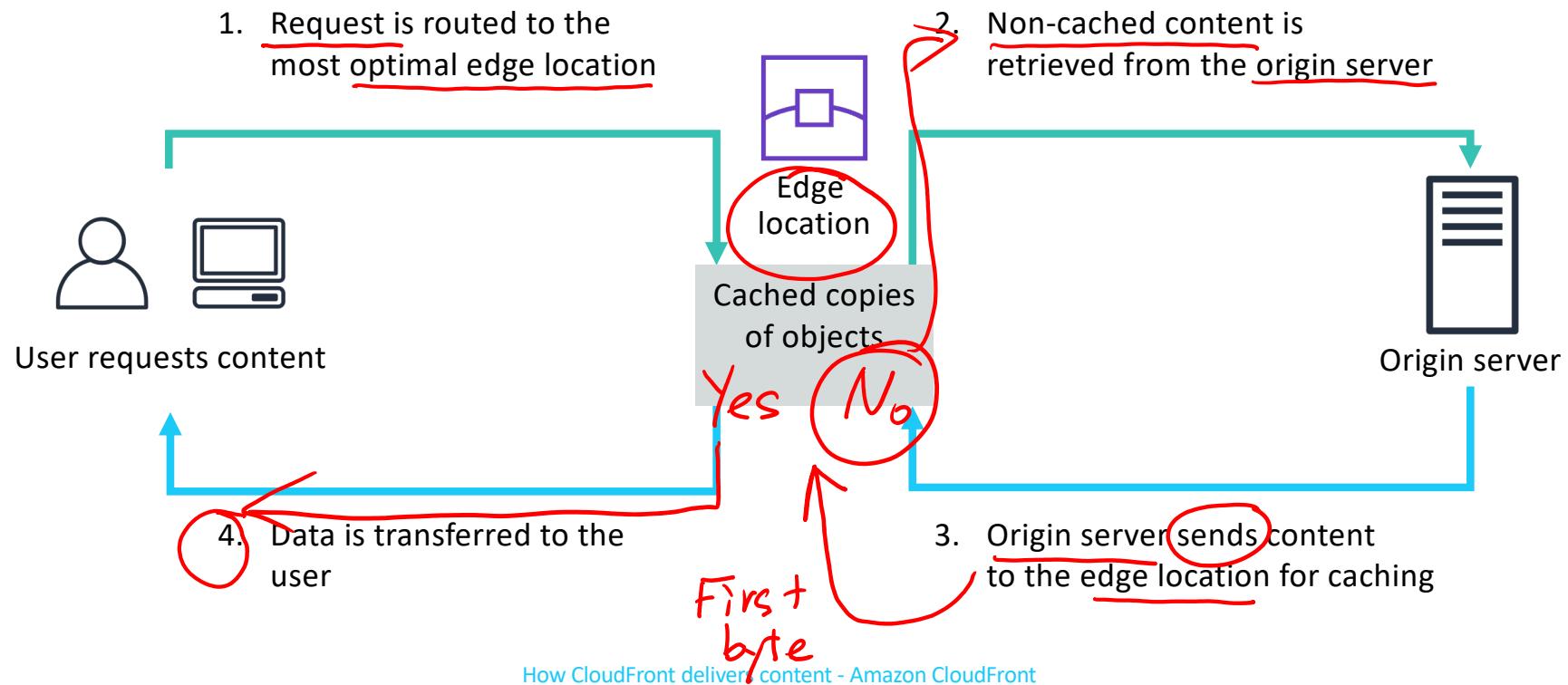
Live Support with Mayday

NEW—Simply tap the "Mayday" button to be connected for free to an Amazon expert who can co-pilot you through any feature by drawing on your screen, walking you through how to do something yourself, or doing it for you—whatever works best. Mayday is available 24x7, 365 days a year, and it's free. Throughout the process, you will be able to see your Amazon Tech advisor live on your screen, but they won't see you. 15 seconds or less is the Mayday response time goal.

Watch it in Action

Kindle Fire HDX is easy to use right out of the box. But when you need extra assistance, the "Mayday" button is there. See how it works in these short commercials.

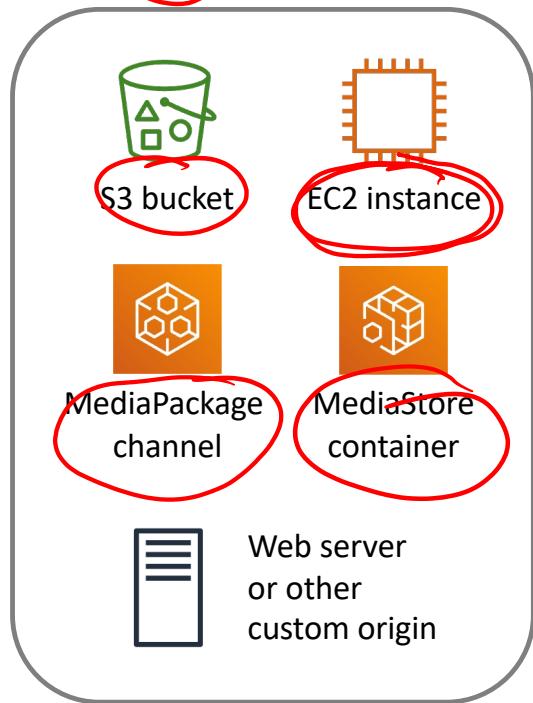
How caching works in Amazon CloudFront



How to configure a CloudFront distribution



1. You specify an origin server



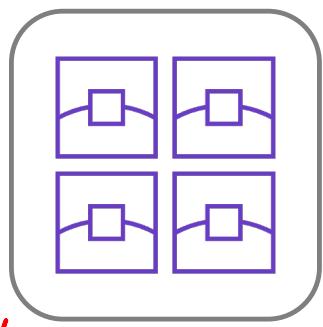
2. You configure the distribution



3. CloudFront assigns a domain name



4. CloudFront sends your distribution's configuration to edge locations



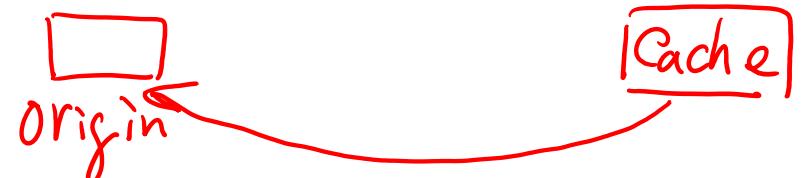
Demo:

<https://www.youtube.com/watch?v=kbI7kRWAU-w>

How to expire content



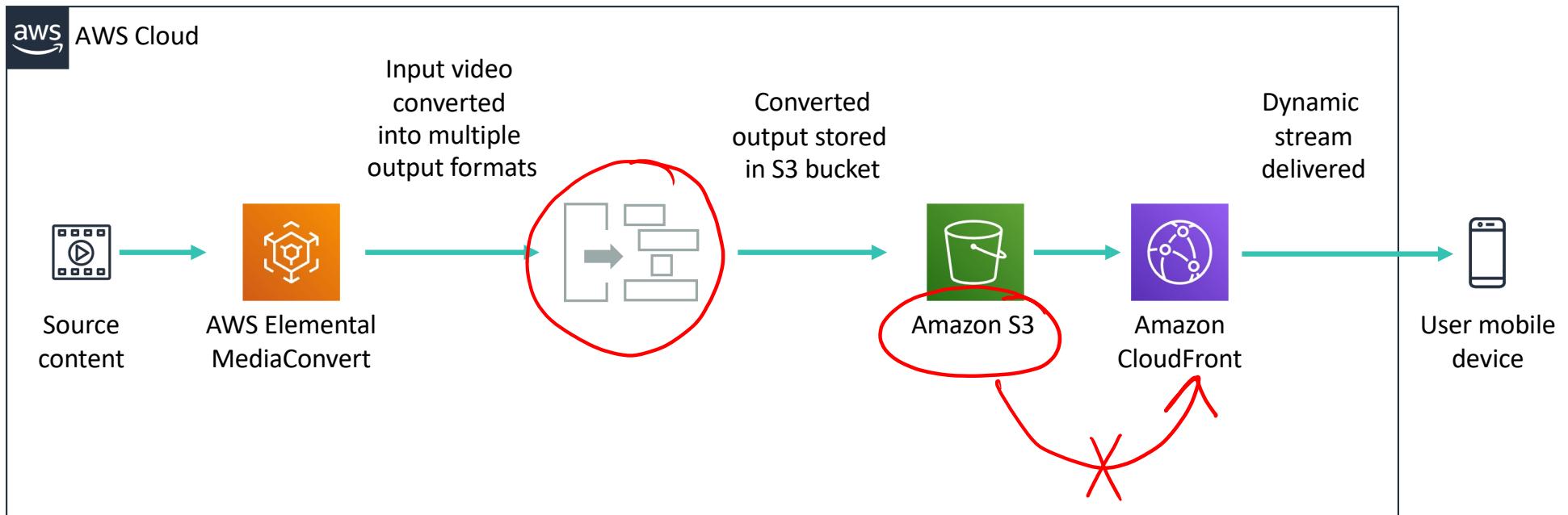
- Time to Live (TTL) –
 - Fixed period of time (expiration period)
 - Set by you $TTL=0$
 - GET request to origin from CloudFront uses If-Modified-Since header
- Change object name –
 - Header v1.jpg becomes Header v2.jpg
 - New name forces immediate refresh
- • Invalidate object –
 - Last resort: inefficient and expensive



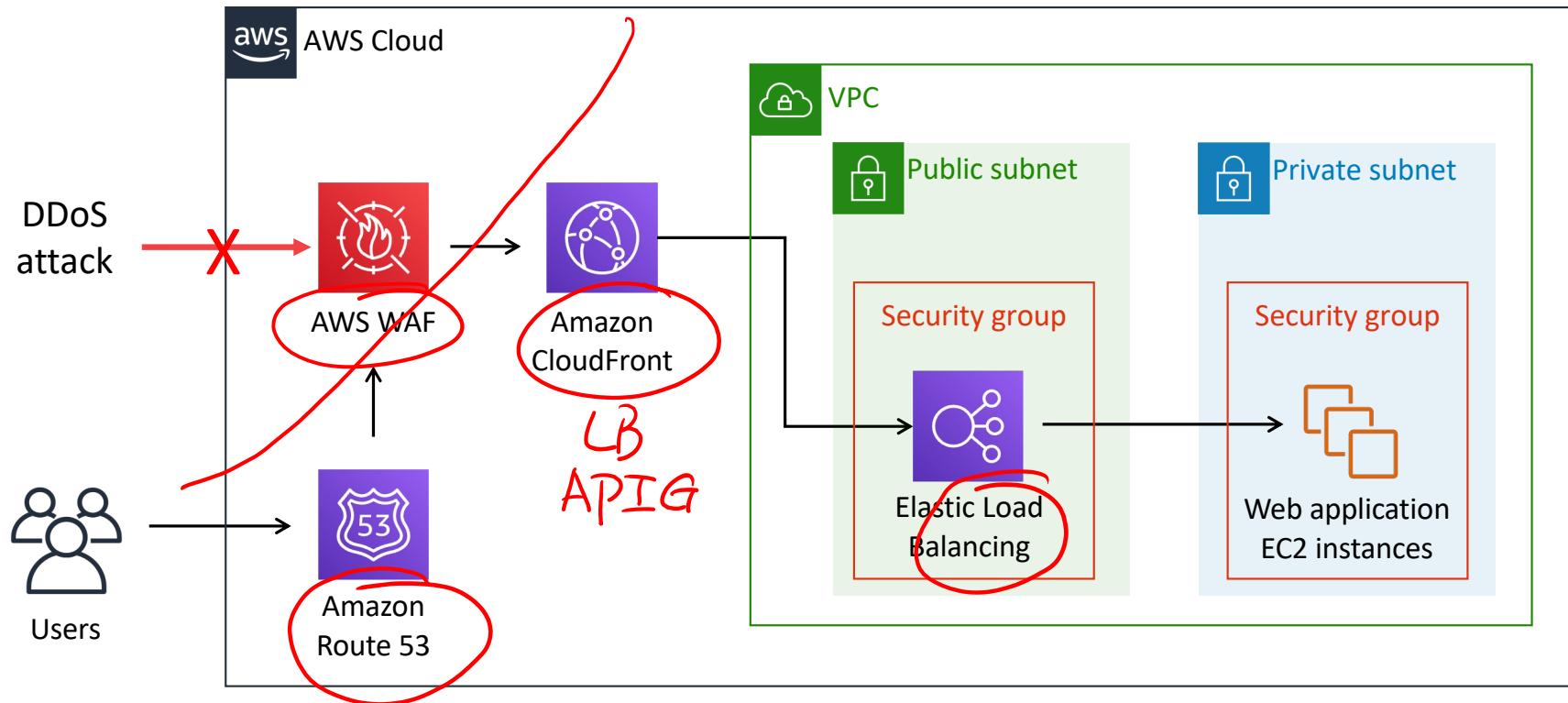
Demo:

https://www.youtube.com/watch?v=qF0q8iRrv_0

Example: Video on demand streaming



Example: DDoS mitigation



Section 3 key takeaways



48



- Amazon CloudFront is a **global CDN service** that accelerates the delivery of content, including static and video, to users with no minimum usage commitments.
- CloudFront uses a global network that comprises **edge locations** and **regional edge caches** to deliver content to your users.
- To use CloudFront to deliver your content, you specify an **origin server** and configure a CloudFront **distribution**. CloudFront assigns a domain name and sends your distribution's configuration to all of its edge locations.
- You can use Amazon CloudFront to **improve the resilience** of your applications that run on AWS from DDoS attacks.