

Amazon CloudFront

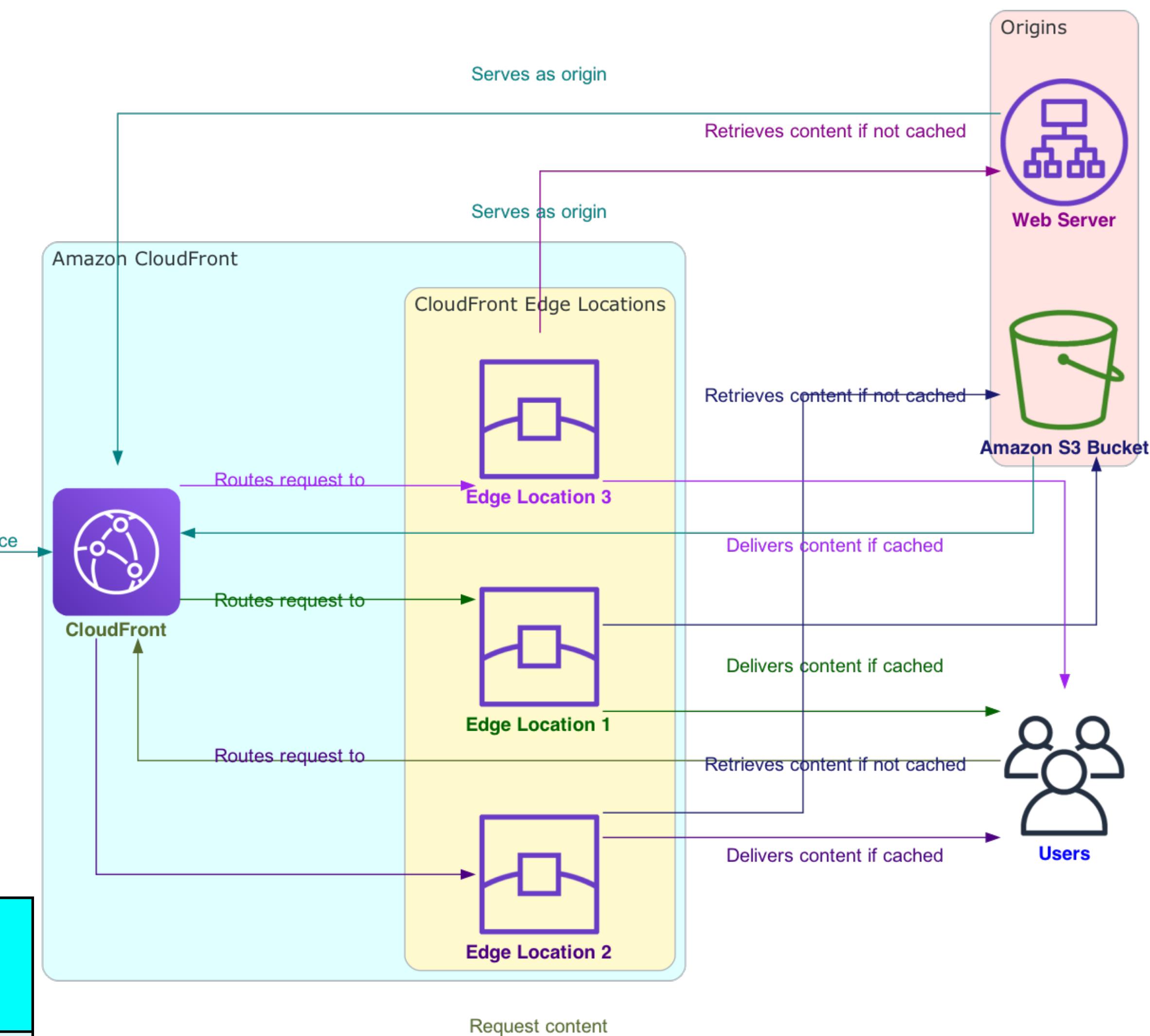
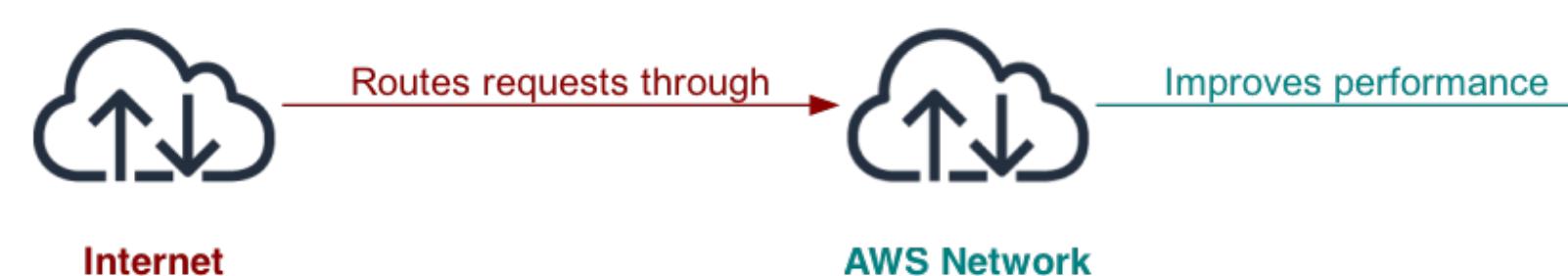
Table of Contents



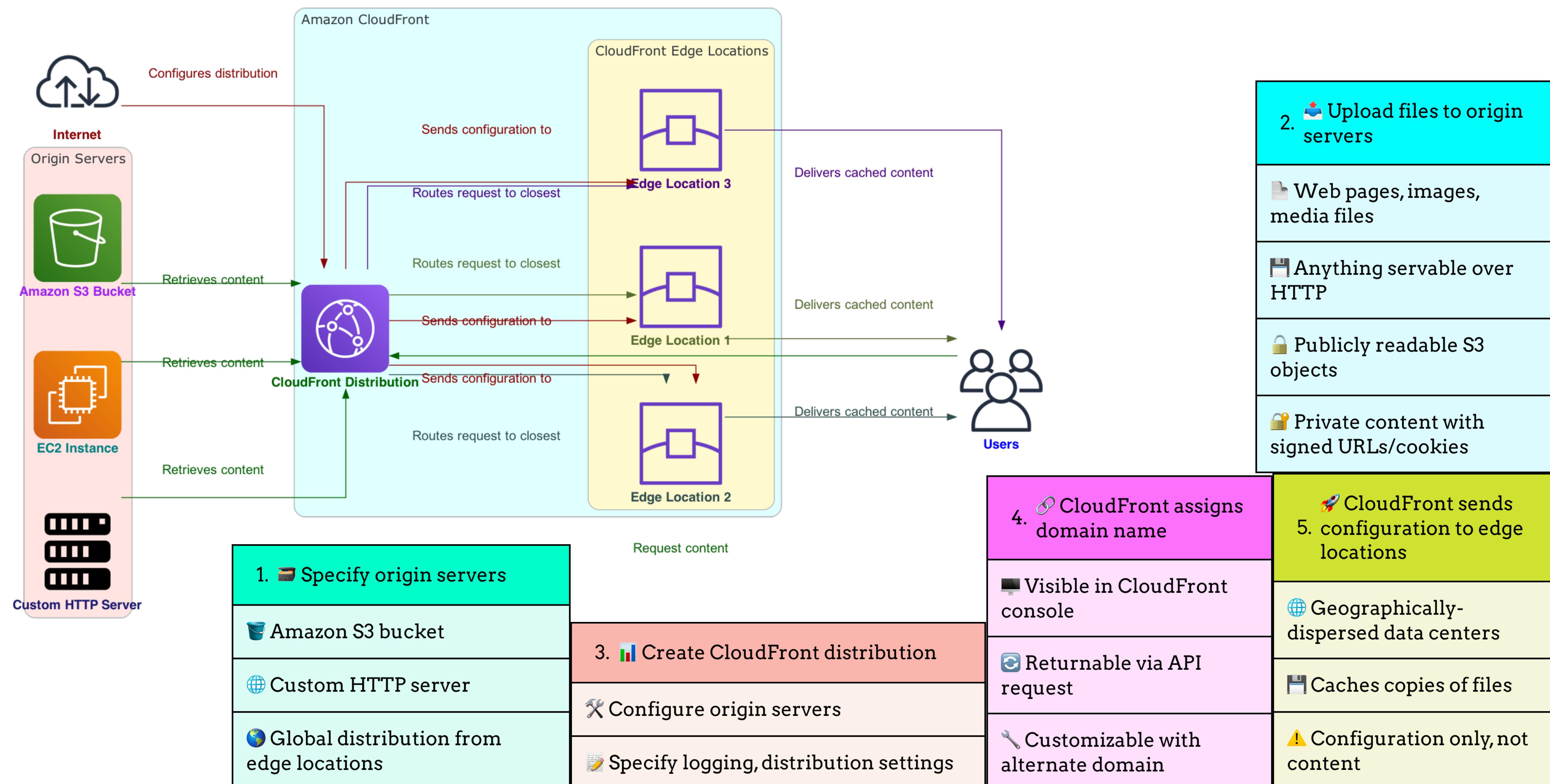
- 1. What is Amazon CloudFront?
- 2. How you configure CloudFront to deliver your content
- 3. Use Cases
- 4. How CloudFront delivers content to your users
- 5. Regional Edge Caches
- 6. Overview of Distributions
- 7. Customizing CloudFront with Policies
- 8. Protecting CloudFront with AWS WAF
- 9. Using HTTPS with CloudFront
- 10. Customizing CloudFront with Edge Functions

What is Amazon CloudFront?

1. 🚀 Speeds up web content distribution	2. 🌎 Global network of edge locations	3. ⏱ Routes requests to lowest latency edge location
📄 Static and dynamic content	🌐 Worldwide data centers	🗺 User request routing
⚡ Faster access for users	🎯 Efficient content delivery	🏆 Best possible performance
4. ⚡ Immediate delivery if content cached		
Lowest latency edge location		
Instant user delivery		
5. 📦 Retrieves content from origins if not cached	6. 🧠 Utilizes AWS backbone network	7. ⏪ Caches content in multiple edge locations
Amazon S3, MediaPackage, HTTP server	Optimal request routing	🌐 Worldwide caching
🔍 Fetches from defined origin	🚀 Improved performance	🔒 Increased reliability and availability

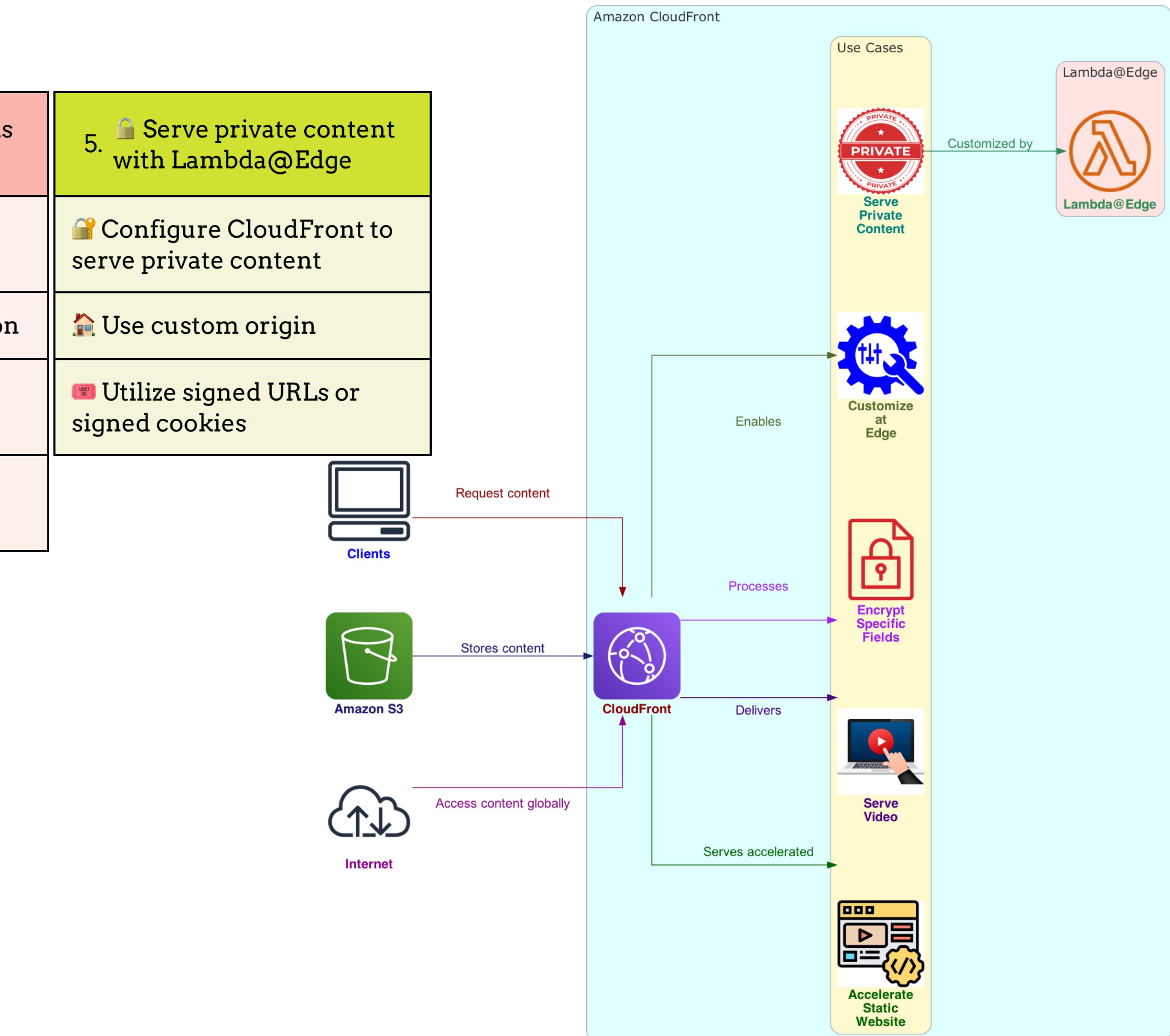


How you configure CloudFront to deliver your content

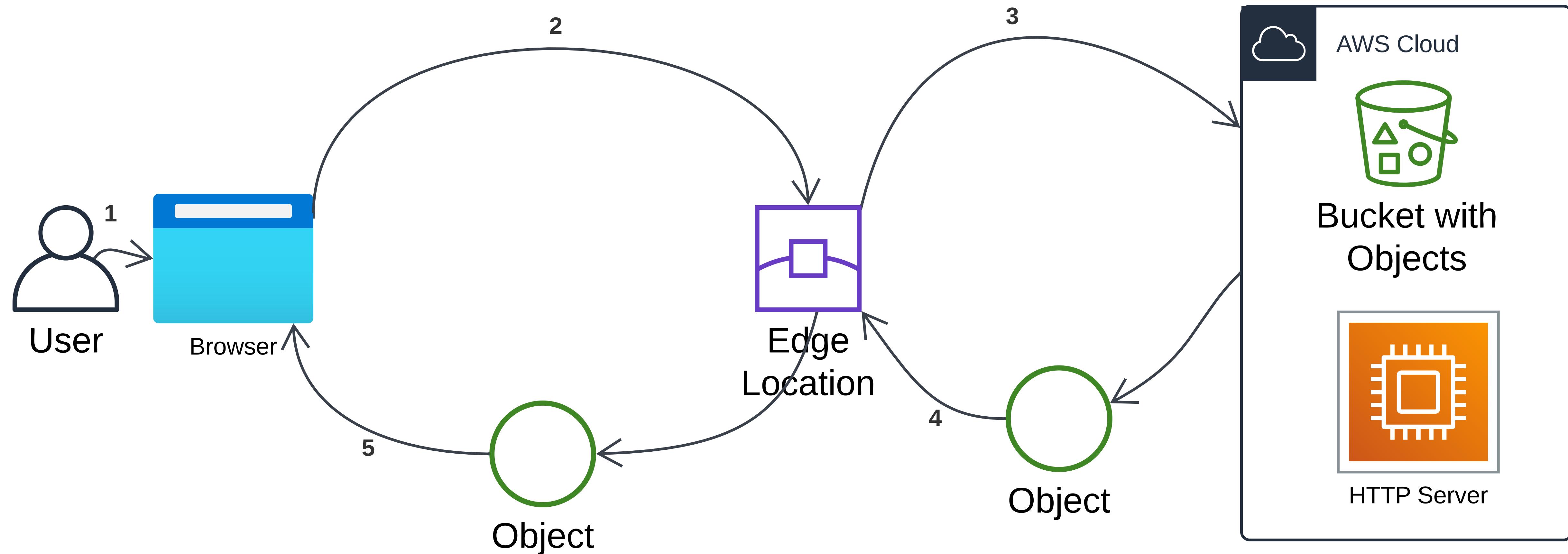


Use Cases

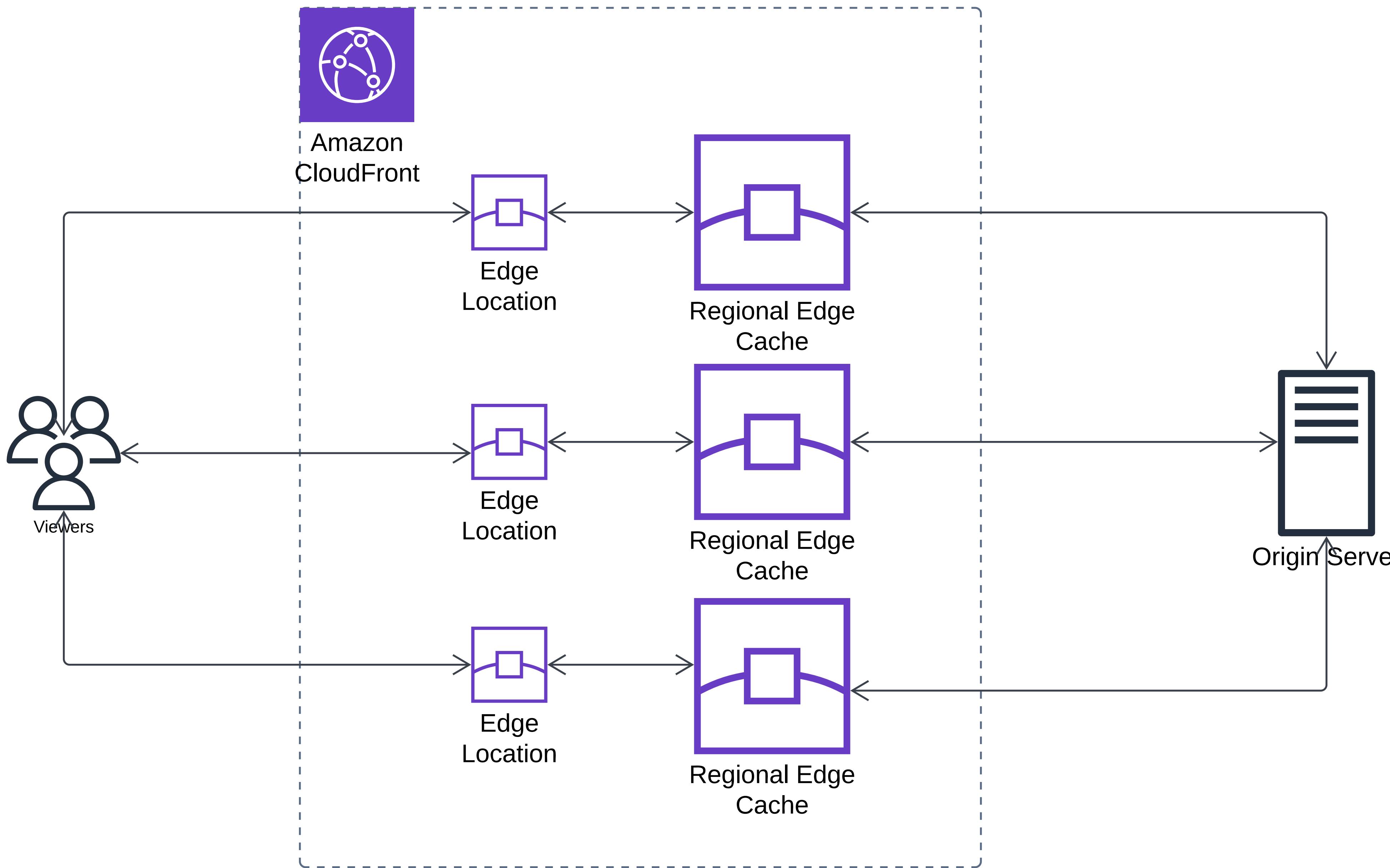
1. 🚀 Accelerate static website content delivery	3. 🔒 Encrypt specific fields throughout processing	5. 🔒 Serve private content with Lambda@Edge
⚡ Speed up static content delivery	🔒 Secure end-to-end connections with HTTPS	🔒 Configure CloudFront to serve private content
🌐 Leverage AWS backbone network, edge servers	🔑 Add field-level encryption	🏡 Use custom origin
🕸️ Enhance user experience	🛡️ Protect data throughout system processing	📅 Utilize signed URLs or signed cookies
2. 📺 Serve video on demand or live streaming	🔍 Restrict data access to specific origin applications	
🎥 Stream pre-recorded files, live events	4. 🛠️ Customize at the edge	
📼 Support common formats (MPEG DASH, HLS, etc.)	💻 Run serverless code at the edge	
📡 Cache media fragments at the edge	⚡ Reduce latency	
🎬 Reduce origin server load	✗ Return custom error messages	
	🔑 Authorize users, control content access	



How CloudFront delivers content to your users



Regional Edge Caches

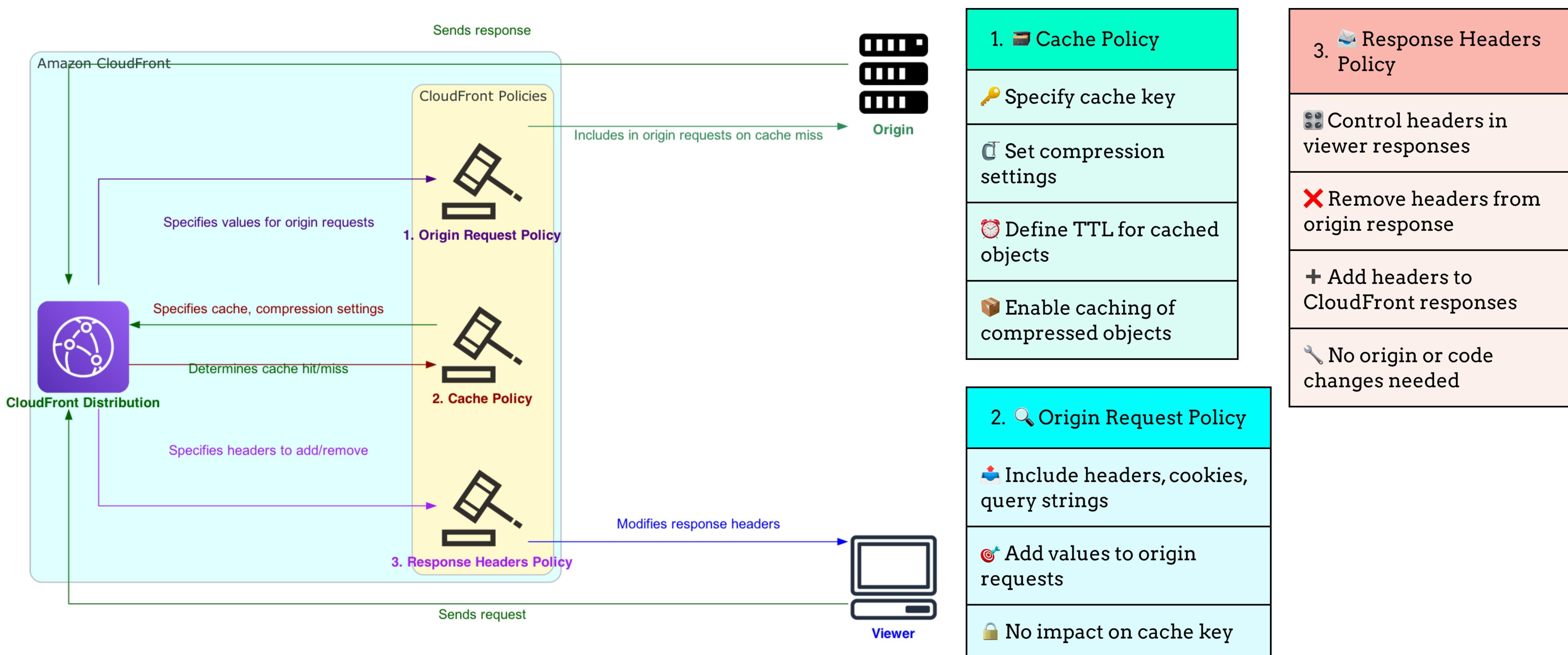


Overview of Distributions

1. ⚙ Purpose	2. 📁 Content Origins (up to 25)	3. 🔑 Access Control	4. 🛡 Security	5. 🗝 Cache Key
Specify content delivery location	Amazon S3 bucket	Available to everyone	Enable AWS WAF protection	Specify values to include
Manage content delivery	AWS Elemental MediaPackage channel AWS Elemental MediaStore container Elastic Load Balancing load balancer HTTP server	Restricted to some users	Require HTTPS for content access	Uniquely identifies cached files
CloudFront Distribution				6. 📱 Origin Request Settings
				Include HTTP headers, cookies, query strings
				7. 🌎 Geographic Restrictions
				Prevent access from selected countries

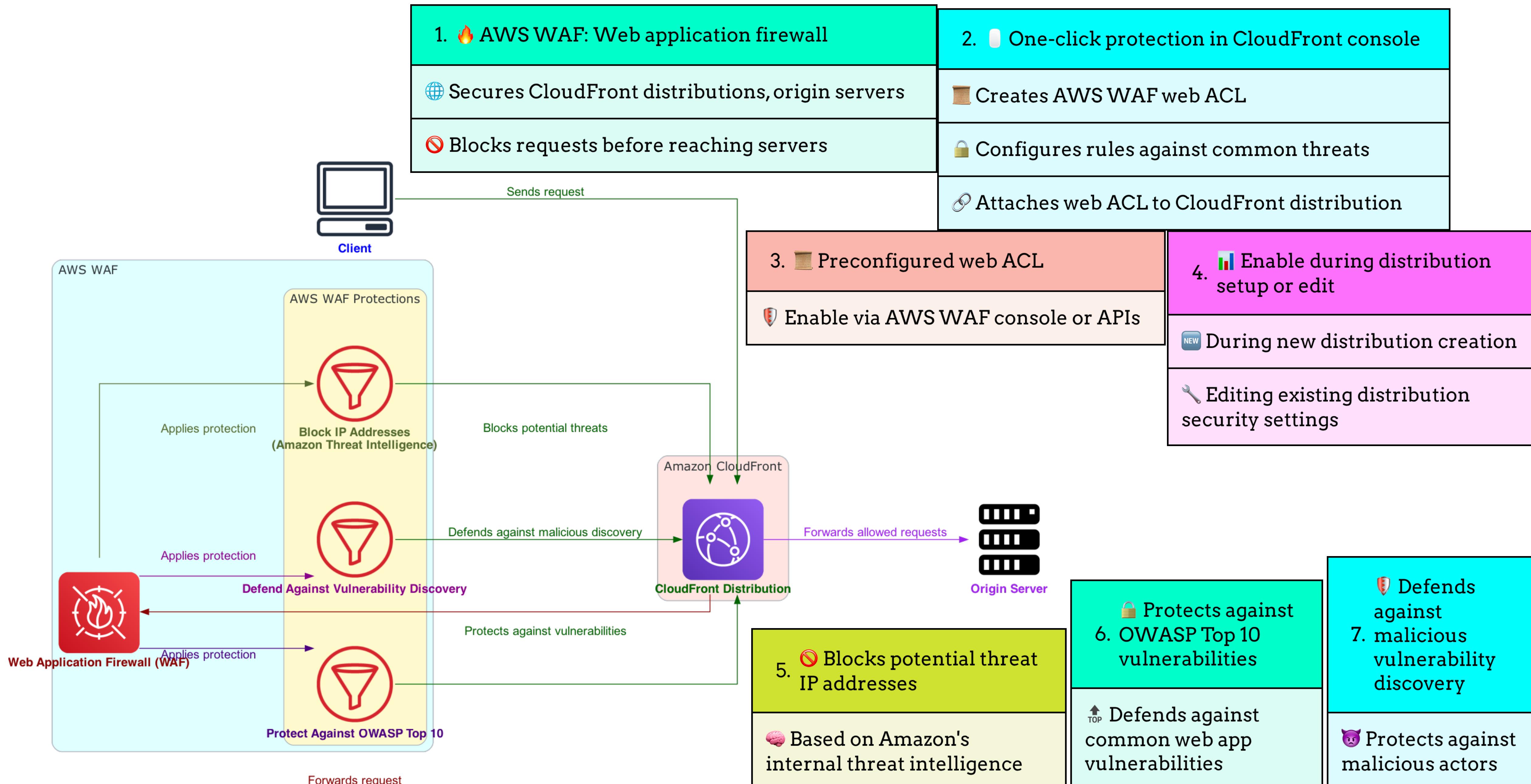


Customizing CloudFront with Policies

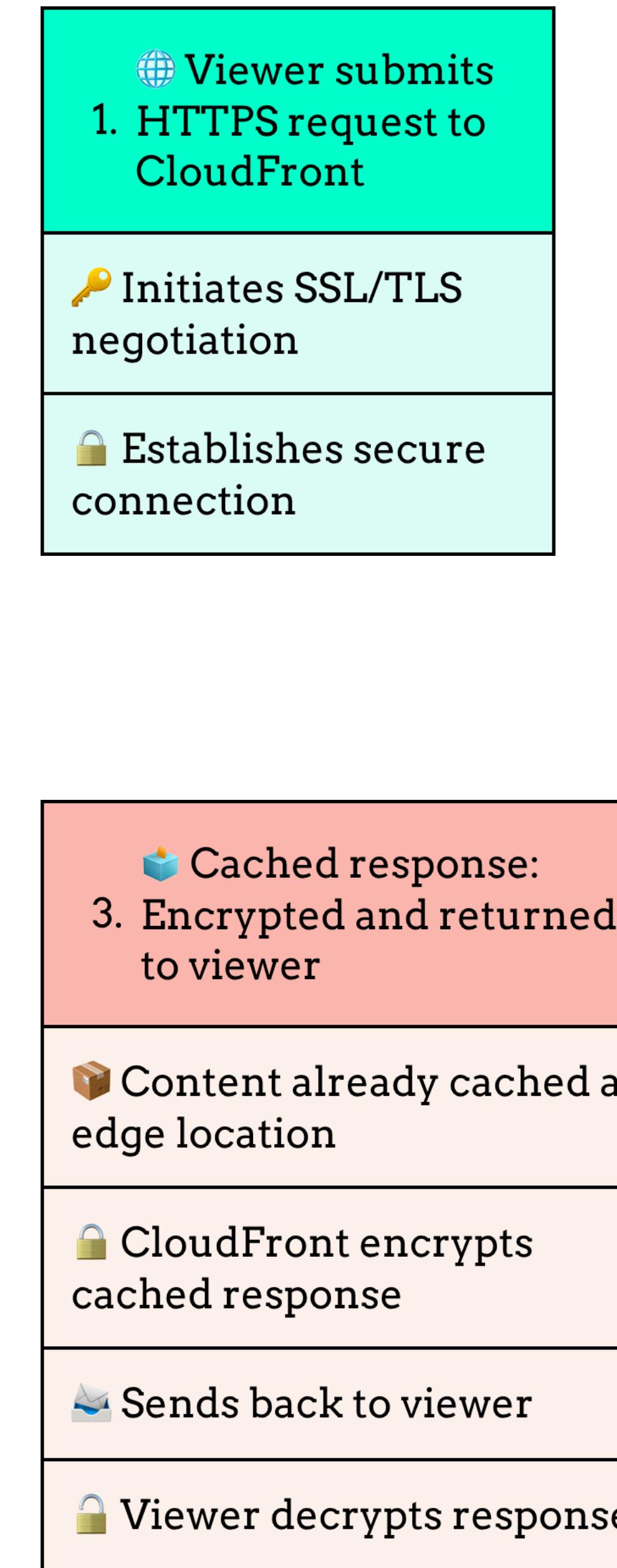
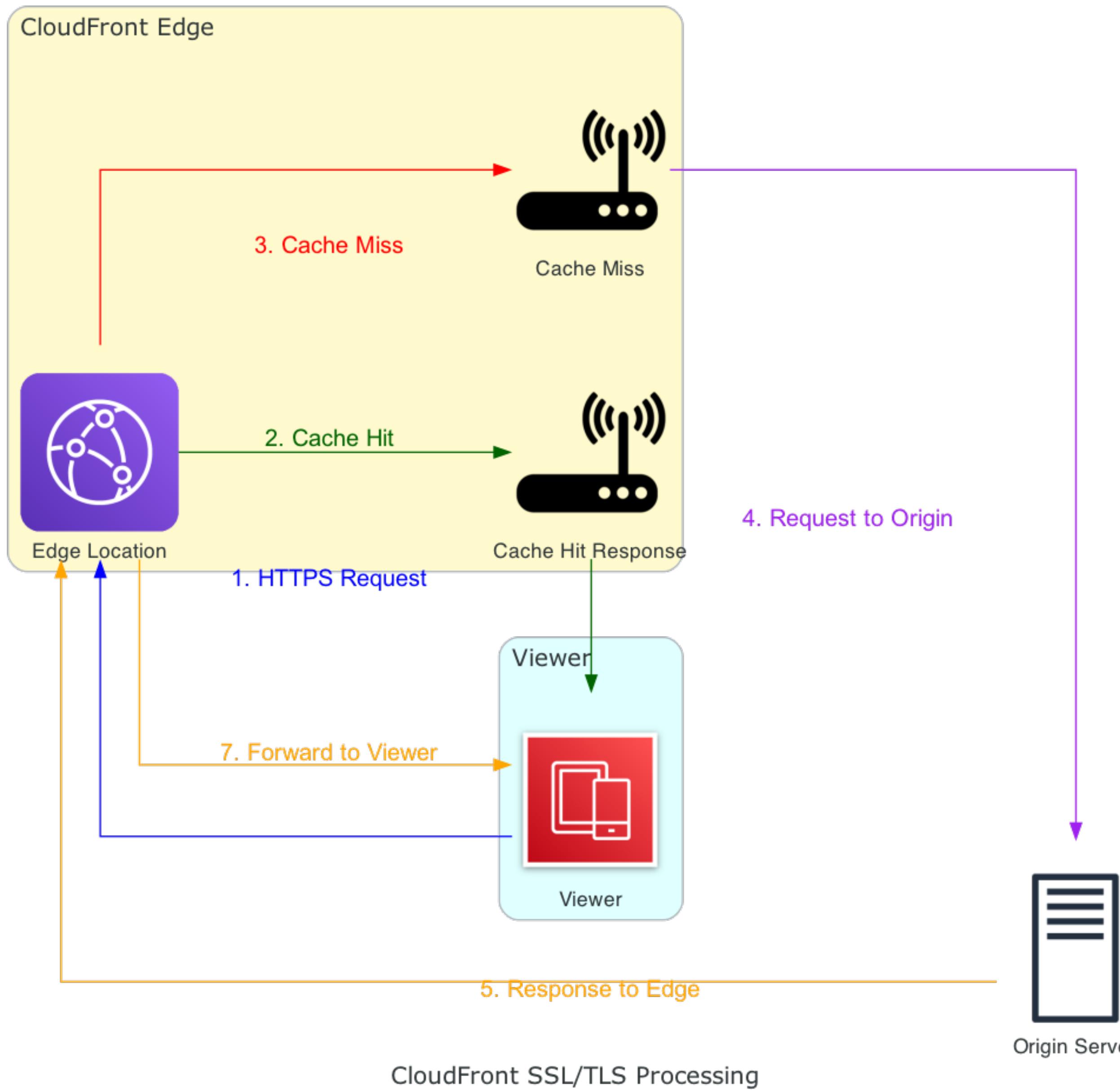




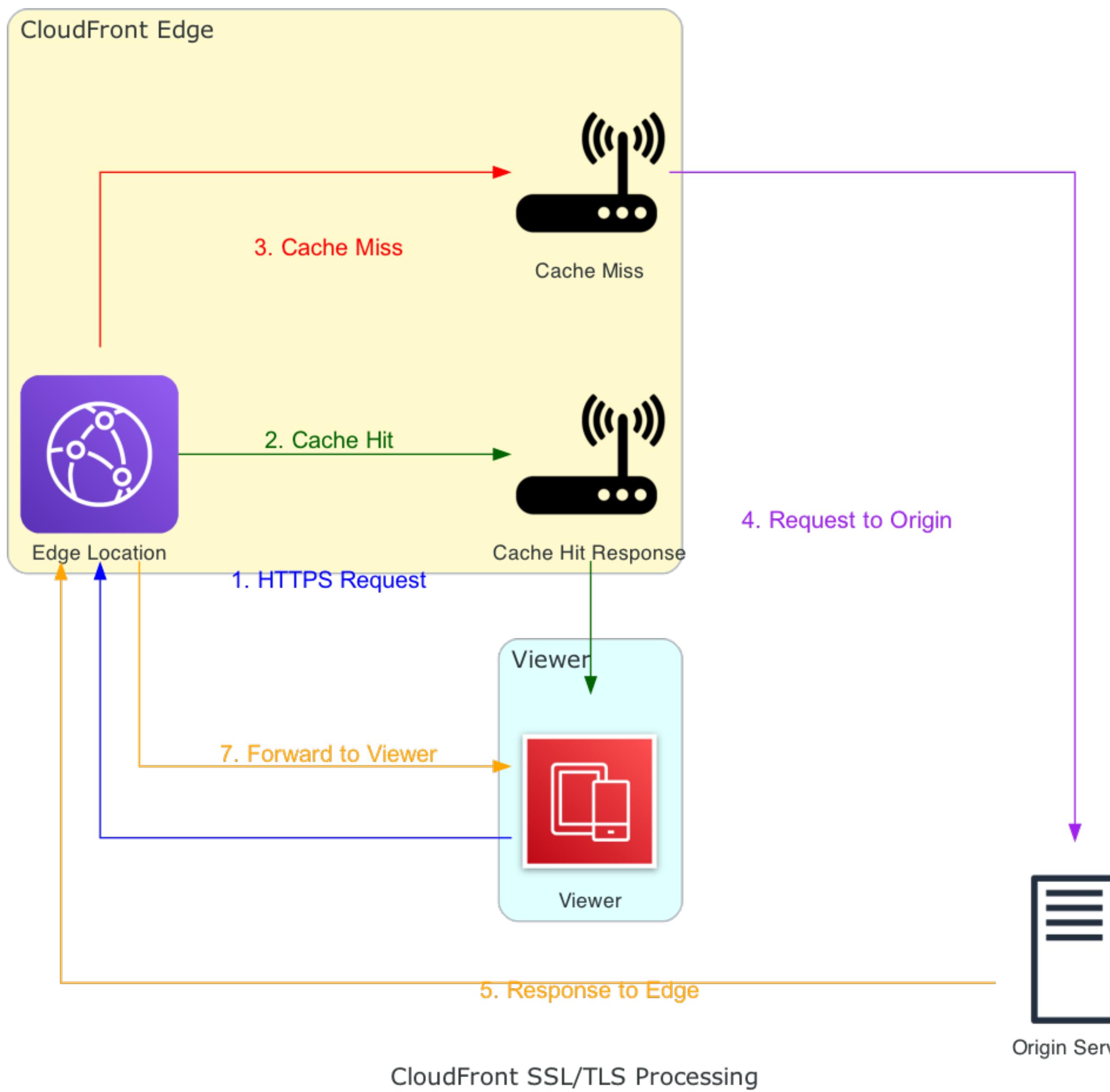
Protecting CloudFront with AWS WAF



Using HTTPS with CloudFront



Using HTTPS with CloudFront



- 5. processes, encrypts response
- Decrypts the request
- Processes the request (generates response)
- Encrypts the response
- Sends back to CloudFront

- 7. Response cached at edge for future requests
- Caches response at edge location
- Future requests served from cache
- Improves performance
- Reduces load on origin server



Customizing CloudFront with Edge Functions

1. Write custom code

Process HTTP requests, responses

Powerful customization options

2. Code runs close to users

Minimal latency

No server management required

3. Manipulate requests, responses

Generate HTTP responses at the edge

Perform authentication, authorization

Perform authentication, authorization

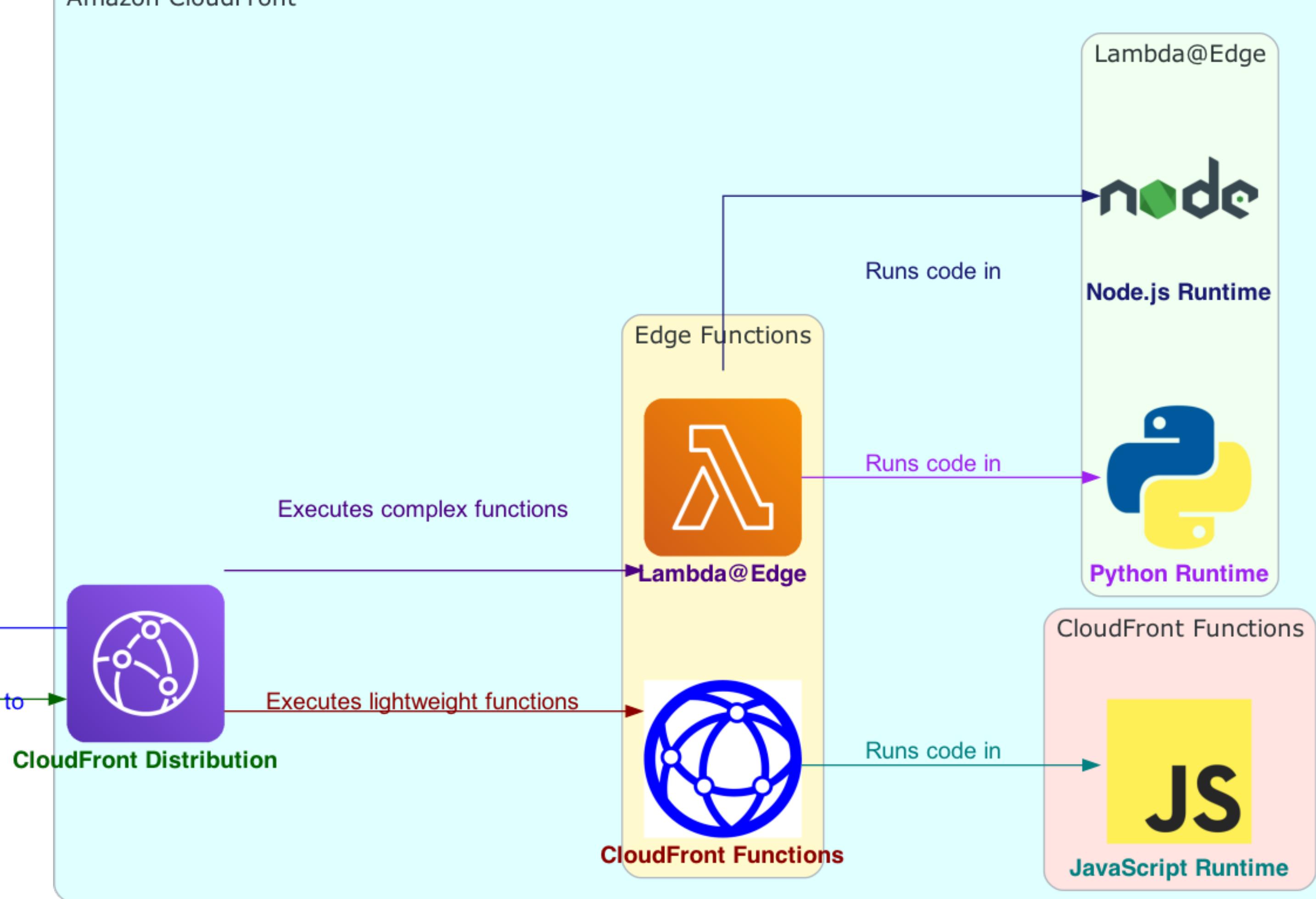
Secure access to content, resources

5. Two options for edge functions

CloudFront Functions: lightweight, high-scale

Lambda@Edge: complex, full application logic

Amazon CloudFront





**Thanks
for
Watching**