1. AWS CloudFront

Purpose:
Amazon CloudFront is a Content Delivery Network (CDN) service that securely delivers data, videos, applications, and APIs to users globally with low latency. It uses a distributed network of edge locations to cache and serve content closer to the user.

Use Case:

- Speeding up the delivery of static and dynamic content (e.g., HTML, CSS, JavaScript, images).
- Streaming video on demand and live.
- Securing applications through SSL/TLS, DDoS protection, and integration with AWS WAF (Web Application Firewall).
- API acceleration, making it faster for users to interact with APIs globally.

State:

- **Cache hit**: Content is found in the CloudFront edge location and served directly.
- **Cache miss**: Content is fetched from the origin (e.g., S3, EC2, or on-premise server) when it's not available in the cache and then cached for future requests.

Task:

- **Action**: CloudFront delivers content via edge locations close to the end-user.
- **Result**: Faster load times and reduced latency, ensuring better user experience and scalability.

Ports:

- **HTTP**: Port **80**
- **HTTPS**: Port **443**

Interrelated Services:

- **AWS S3**: Often used as the origin for static content.
- **AWS Lambda@Edge**: Allows for the execution of custom logic at CloudFront edge locations.
- **AWS WAF**: Works with CloudFront to protect web applications from common web exploits.
- **AWS Shield**: Provides DDoS protection for CloudFront distributions.

2. Redis (Amazon ElastiCache for Redis)

Purpose:
Redis is an in-memory key-value store that can be used as a database, cache, or message broker. Amazon ElastiCache for Redis provides a managed service for Redis, offering improved performance, scalability, and availability.

Use Case:

- **Caching**: Storing frequently accessed data (e.g., session data, user profiles) to reduce the load on databases and provide faster responses.
- **Real-time analytics**: In-memory data processing for real-time dashboards and data insights.
- **Message queues**: Utilizing Redis Pub/Sub for event streaming or inter-service communication.
- **Leaderboard/Counting**: In scenarios like gaming leaderboards or live event counting, Redis is often used because of its high-speed write/read operations.

State:

- **Volatile**: Redis stores data in memory, making it volatile unless persistence is explicitly enabled.
- **Non-persistent**: By default, Redis doesn't persist data to disk. However, it supports snapshots (RDB) or Append Only File (AOF) for persistence.

Task:

- **Action**: Redis handles data in-memory for extremely low latency data access.
- **Result**: Rapid data access, often improving application performance by reducing the load on relational databases.

Ports:

- **Default Redis Port**: Port **6379**
- **With TLS/SSL**: Port **6380** (if configured for encryption)

Interrelated Services:

- **Amazon EC2**: Applications running on EC2 can integrate with Redis for caching and data store needs.
- **Amazon RDS**: Redis can be used to cache results from RDS databases to improve performance.
- **Elastic Load Balancing (ELB)**: While Redis does not directly interface with ELB, it can be used in tandem with it to scale application front ends, while Redis handles backend caching.

Key Differences:

Feature	AWS CloudFront	Redis (ElastiCache for Redis)
Primary Purpose	Content delivery network (CDN)	In-memory data store and cache
State	Non-persistent caching of content	In-memory, optionally persistent
Use Case	Delivering web content, video, API acceleration	Caching, session store, real-time analytics
Port Number	HTTP (80), HTTPS (443)	Default (6379), SSL (6380)
Interrelated Services	S3, Lambda@Edge, WAF, Shield, etc.	EC2, RDS, Lambda, etc.

Both services have different use cases but can be used together to optimize web applications. For example, you can use Redis to cache application data and CloudFront to cache static and dynamic content, ensuring a fast and scalable web experience.