**Src:** https://letstalkaws.com/thread-59.html

A few days ago someone asked what is the difference between these services as their functionality looks very similar on the surface. Below is a short write up I did to help bring some clarity for those who are still new to AWS.  
  
**1. Route53 Geolocation Policy**  
  
This routing policy is specifically used if you always want users from a certain location i.e a country, continent or in the case of US, a specific state to be always given an IP of the same environment/region, everytime. This is used for content localization, for example if you want people from Europe to only access an environment/region that hosts content that is local/relevant to that region. Think of this like when you open youtube from different parts in the world, you see content related to that region on the home page.  
  
![A diagram of a computer program

Description automatically generated]()     
  
This means that unless you have setup a failover route as well for the same FQDN using traffic policy inside Route53, then if that region fails, the users will get failure screens with 4xx errors. Also, if your DNS record TTL values are high i.e >300, then there is a possibility that even though your environment might have recovered from a failure using new endpoint IPs, existing users who might already have a failed environment IP will still get the failure as DNS changes can take a long time to propogate fully.  
  
  
**2. Cloudfront**  
  
This is specifically used only for speeding up the delivery of your content. It caches your content on the global edge locations, and it does not route repeated requests to your environment, which means if you make changes, then you have to wait till you invalidate the cache across the globe for the new content to replace old one. Also, the source is a single point, so in cases of that environment failing or a region failing, no new content can be delivered to users while your env is down. The advantage is ofcourse that you do not need to run high capacity servers because lesser requests are hitting your origin servers or lesser API requests hitting your S3 buckets. CloudFront supports only HTTP & HTTPS requests.  
  
![A screenshot of a computer

Description automatically generated]()     
  
  
**3. Global accelerator**  
  
This service does not cache your content like CloudFront. It basically wants your users from anywhere in the world to jump onto the AWS network at their closest point and then get shot through the AWS global network to reach the closest environment/region where you have running resources. This is much faster than traversing the normal submarine cables which offer no guaranteed QoS and are plagued by bandwidth throttling courtesy telcos. The only similarity with CloudFront is that it makes use of edge locations to let your customers into the AWS global network.  
  
Unlike cloudfront, you can use a wide range of TCP & UDP listener ports and you can have multiple regions behind your GlobalAccelerator which means higher availability. Also, since it provides anycast IPs, all your environments regardless of region and number of environments will have 2 global IPs which means any third party provider integration, you do not need to be hassled with multiple IPs and also whitelisting on firewalls by others becomes a breeze. Also, failovers are much more quicker since there is no DNS propagation to wait for. All in all, this will greatly reduce latency without need for caching any content. Super useful for use-cases like live gaming.  
  
A map of the world

Description automatically generated  
  
Global Accelerator Image source: [https://aws.amazon.com/blogs/networking-...celerator/](https://aws.amazon.com/blogs/networking-and-content-delivery/traffic-management-with-aws-global-accelerator/)