AWS Whitepaper

AWS Best Practices for DDoS Resiliency

AWS Best Practices for DDoS Resiliency: AWS Whitepaper

Table of Contents

AWS Best Practices for DDoS Resiliency

Publication date: **August 9, 2023** (*Document revisions*)

It's important to protect your business from the impact of Distributed Denial of Service (DDoS) attacks, as well as other cyberattacks. Keeping customer trust in your service by maintaining the availability and responsiveness of your application is high priority. You also want to avoid unnecessary direct costs when your infrastructure must scale in response to an attack. Amazon Web Services (AWS) is committed to providing you with the tools, best practices, and services to defend against bad actors on the internet. Using the right services from AWS helps ensure high availability, security, and resiliency.

In this whitepaper, AWS provides you with prescriptive DDoS guidance to improve the resiliency of applications running on AWS. This includes a DDoS-resilient reference architecture that can be used as a guide to help protect application availability. This whitepaper also describes different attack types, such as infrastructure layer attacks and application layer attacks. AWS explains which best practices are most effective to manage each attack type. In addition, the services and features that fit into a DDoS mitigation strategy are outlined, along with how each one can be used to help protect your applications.

This paper is intended for IT decision makers and security engineers who are familiar with the basic concepts of networking, security, and AWS. Each section has links to AWS documentation that provides more detail on the best practice or capability.

AWS detects over a million DDoS attacks per year and mitigates thousands on a daily basis against our customers. According to our Shield Response team (SRT), the majority of customers who experience business impact from DDoS attacks have not implemented the recommendations in this guide.

Are you Well-Architected?

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems in the cloud. The six pillars of the Framework allow you to learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. Using the AWS Well-Architected Tool, available at no charge in the AWS Management Console (sign-in required), you can review your workloads against these best practices by answering a set of questions for each pillar.

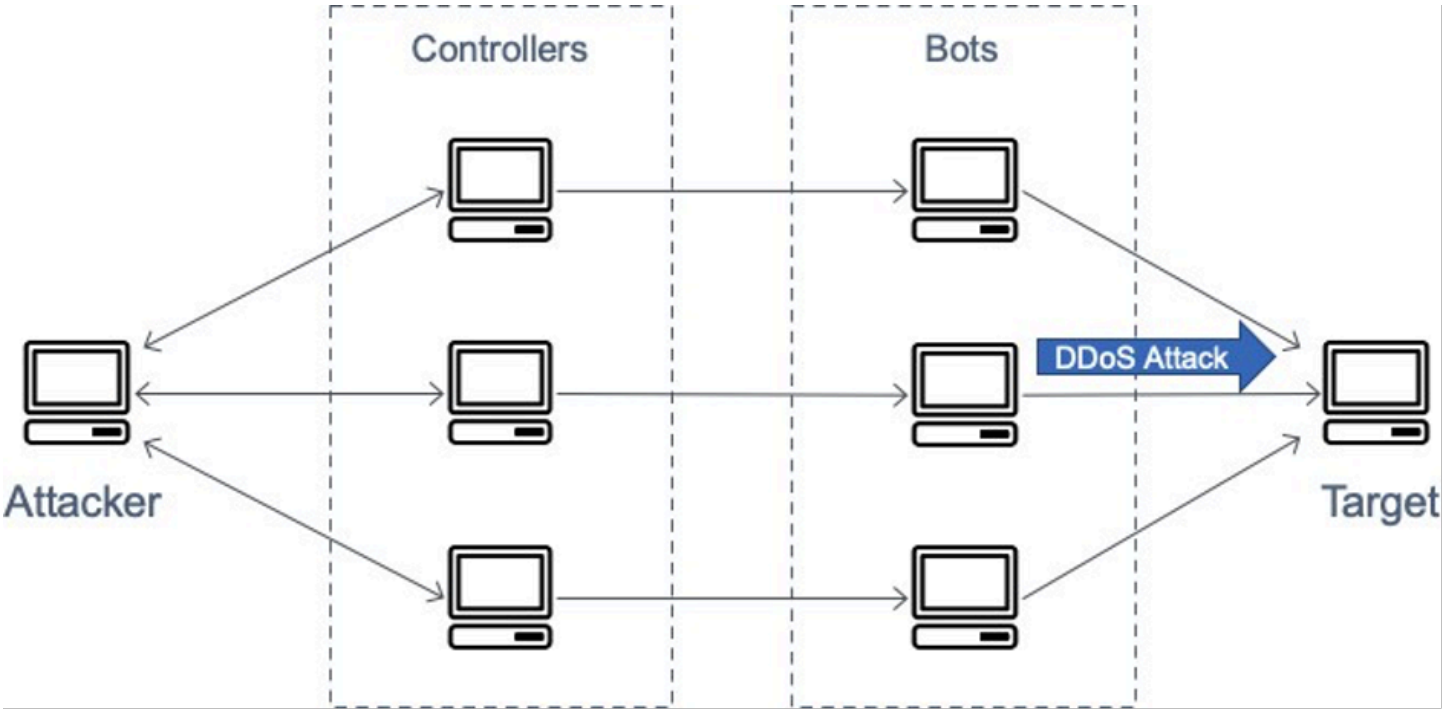
For more expert guidance and best practices for your cloud architecture—reference architecture deployments, diagrams, and whitepapers, refer to the AWS Architecture Center.

Introduction to denial of service attacks

A Denial of Service (DoS) attack, or event, is a deliberate attempt to make a website or application unavailable to users, such as by flooding it with network traffic. Attackers use a variety of techniques that consume large amounts of network bandwidth or tie up other system resources, disrupting access for legitimate users. In its simplest form, a lone attacker uses a single source to carry out a DoS attack against a target, as shown in the following figure.

A diagram depicting a DoS attack

In a Distributed Denial of Service (DDoS) attack, an attacker uses multiple sources to orchestrate an attack against a target. These sources can include distributed groups of malware infected computers, routers, IoT devices, and other endpoints. The following figure shows a network of compromised hosts that participate in the attack, generating a flood of packets or requests to overwhelm the target.



A diagram depicting a DDoS attack

There are seven layers in the Open Systems Interconnection (OSI) model, and they are described in the following table. DDoS attacks are most common at layers 3, 4, 6, and 7.

- Layer 3 and 4 attacks correspond to the Network and Transport layers of the OSI model. Within this whitepaper, AWS refers to these collectively as infrastructure layer attacks.
- Layer 6 and 7 attacks correspond to the Presentation and Application layers of the OSI model. This whitepaper addresses these together as application layer attacks.

This paper discusses these attack types in the sections that follow.

Table 1 — OSI model

#	Layer	Unit	Description	Vector examples
7	Application	Data	Network process to application	HTTP floods, DNS query floods
6	Presentation	Data	Data represent ation and encryption	Transport Layer Security (TLS) abuse
5	Session	Data	Interhost communication	N/A
4	Transport	Segments	End-to-end connections and reliability	Synchronize (SYN) floods
3	Network	Packets	Path determina tion and logical addressing	User Datagram Protocol (UDP) reflection attacks
2	Data Link	Frames	Physical addressing	N/A

#	Layer	Unit	Description	Vector examples
1	Physical	Bits	Media, signal, and binary transmission	N/A

Infrastructure layer attacks

The most common DDoS attacks, User Datagram Protocol (UDP) reflection attacks and SYN floods, are *infrastructure layer attacks*. An attacker can use either of these methods to generate large volumes of traffic that can inundate the capacity of a network or tie up resources on systems such as servers, firewalls, intrusion prevention system (IPS), or load balancer. While these attacks can be easy to identify, to mitigate them effectively, you must have a network or systems that scale up capacity more rapidly than the inbound traffic flood. This extra capacity is necessary to either filter out or absorb the attack traffic freeing up the system and application to respond to legitimate customer traffic.

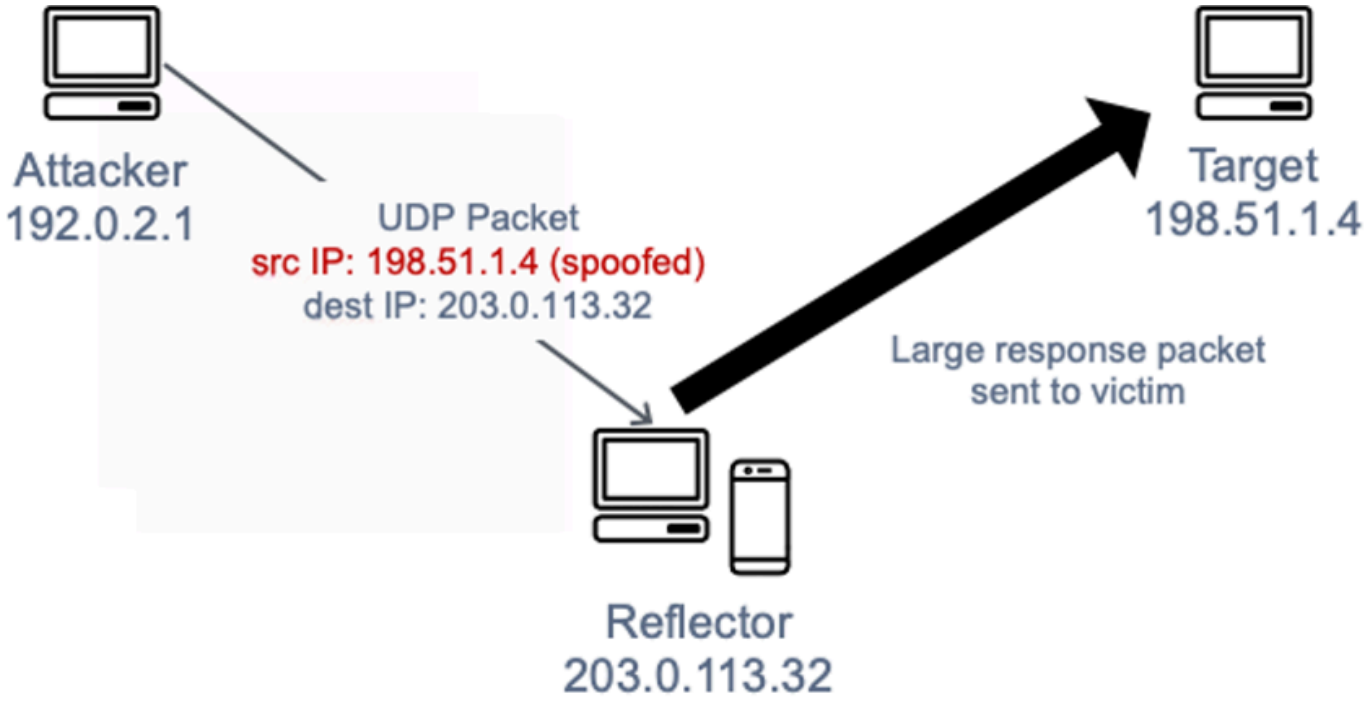
UDP reflection attacks

UDP reflection attacks exploit the fact that UDP is a stateless protocol. Attackers can craft a valid UDP request packet listing the attack target's IP address as the UDP source IP address. The attacker has now falsified—spoofed—the UDP request packet's source IP. The UDP packet contains the spoofed source IP and is sent by the attacker to an intermediate server. The server is tricked into sending its UDP response packets to the targeted victim IP rather than back to the attacker's IP address. The intermediate server is used because it generates a response that is several times larger than the request packet, effectively amplifying the amount of attack traffic sent to the target IP address.

The amplification factor is the ratio of response size to request size, and it varies depending on which protocol the attacker uses: DNS, Network Time Protocol (NTP), Simple Service Directory Protocol (SSDP), Connectionless Lightweight Directory Access Protocol (CLDAP), Memcached, Character Generator Protocol (CharGen), or Quote of the Day (QOTD).

For example, the amplification factor for DNS can be 28 to 54 times the original number of bytes. So, if an attacker sends a request payload of 64 bytes to a DNS server, they can generate over 3400 bytes of unwanted traffic to an attack target. UDP reflection attacks are accountable for larger

volume of traffic in comparison to other attacks. The following figure illustrates the reflection tactic and amplification effect.

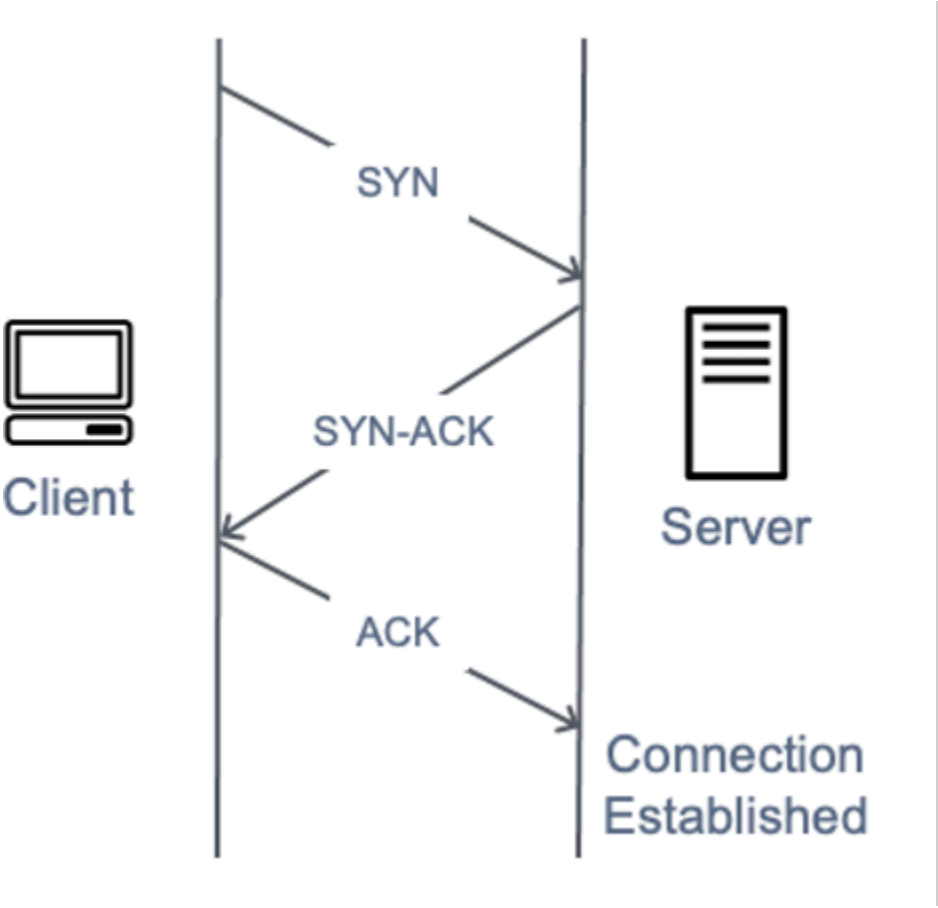


A diagram depicting a UDP reflection attack

It should be noted that reflection attacks, while they provide attackers with "free" amplification, require IP spoofing capability and as increasing numbers of network providers adopt Source Address Validation Everywhere (SAVE) or BCP38, this capability is removed, requiring DDoS service providers cease reflection attacks or to relocate to data centers and network providers who do not implement source address validation.

SYN flood attacks

When a user connects to a Transmission Control Protocol (TCP) service, such as a web server, their client sends a SYN packet. The server returns a synchronization-acknowledgement (SYN-ACK) packet, and finally the client responds with an acknowledgement (ACK) packet, which completes the expected three-way handshake. The following image illustrates this typical handshake.



A diagram depicting a SYN three-way handshake

In a SYN flood attack, a malicious client sends a large number of SYN packets, but never sends the final ACK packets to complete the handshakes. The server is left waiting for a response to the half-open TCP connections and the idea is that the target eventually runs out of capacity to accept new TCP connections which prevents new users from connecting to the server, however the actual impact is more nuanced. Modern operating systems all implement SYN cookies by default as a mechanism to counter state table exhaustion from SYN flood attacks. Once the SYN queue length reaches a pre-determined threshold, the server responds with a SYN-ACK containing a crafted initial sequence number, without creating an entry in its SYN queue. If the server then receives an ACK containing a correctly incremented acknowledgement number it's able to add the entry to its state table and proceed as normal. The actual impact of SYN floods on target devices tends to be network capacity and CPU exhaustion, however intermediate stateful devices such as firewalls (or EC2 security group connection tracking) may suffer TCP state table exhaustion and drop new connections.

TCP middlebox reflection

This relatively new attack vector was first disclosed in an academic whitepaper in August 2021 which explained how TCP non-compliance in both nation-state and commercially available firewalls could result in these being tricked into becoming a TCP amplification vector. We have seen these attacks "in the wild" since early 2022 and continue to see them today. The amplification factor varies due to the different ways in which vendors have implemented this "feature", but can exceed Memcached UDP amplification.

Application layer attacks

An attacker may target the application itself by using a layer 7 or application layer attack. In these attacks, similar to SYN flood infrastructure attacks, the attacker attempts to overload specific functions of an application to make the application unavailable or unresponsive to legitimate users. Sometimes this can be achieved with very low request volumes that generate only a small volume of network traffic. This can make the attack difficult to detect and mitigate. Examples of application layer attacks include HTTP floods, cache-busting attacks, and WordPress XML-RPC floods.

- In an *HTTP flood attack*, an attacker sends HTTP requests that appear to be from a valid user of the web application. Some HTTP floods target a specific resource, while more complex HTTP floods attempt to emulate human interaction with the application. This can increase the difficulty of using common mitigation techniques such as request rate limiting.
- *Cache-busting attacks* are a type of HTTP flood that uses variations in the query string to circumvent content delivery network (CDN) caching. Instead of being able to return cached results, the CDN must contact the origin server for every page request, and these origin fetches cause additional strain on the application web server.
- With a *WordPress XML-RPC flood attack*, also known as a WordPress pingback flood, an attacker targets a website hosted on the WordPress content management software. The attacker misuses the XML-RPC API function to generate a flood of HTTP requests. The pingback feature allows a website hosted on WordPress (Site A) to notify a different WordPress site (Site B) through a link that Site A has created to Site B. Site B then attempts to fetch Site A to verify the existence of the link. In a pingback flood, the attacker misuses this capability to cause Site B to attack Site A. This type of attack has a clear signature: "`WordPress:`" is typically present in the User-Agent of the HTTP request header.

There are other forms of malicious traffic that can impact an application's availability. *Scraper bots* automate attempts to access a web application to steal content or record competitive information, such as pricing. *Brute force* and *credential stuffing* attacks are programmed efforts to gain unauthorized access to secure areas of an application. These are not strictly DDoS attacks, but their automated nature can look similar to a DDoS attack and they can be mitigated by implementing some of the same best practices to be covered in this paper.

Application layer attacks can also target Domain Name System (DNS) services. The most common of these attacks is a *DNS query flood* in which an attacker uses many well-formed DNS queries to exhaust the resources of a DNS server. These attacks can also include a cache-busting component where the attacker randomizes the subdomain string to bypass the local DNS cache of any given resolver. As a result, the resolver can't take advantage of cached domain queries and must instead repeatedly contact the authoritative DNS server, which amplifies the attack.

If a web application is delivered over Transport Layer Security (TLS), an attacker can also choose to attack the TLS negotiation process. TLS is computationally expensive so an attacker, by generating extra workload on the server to process unreadable data (or unintelligible (ciphertext)) as a legitimate handshake, can reduce server's availability. In a variation of this attack, an attacker completes the TLS handshake but perpetually renegotiates the encryption method. An attacker can alternatively attempt to exhaust server resources by opening and closing many TLS sessions.

Mitigation techniques

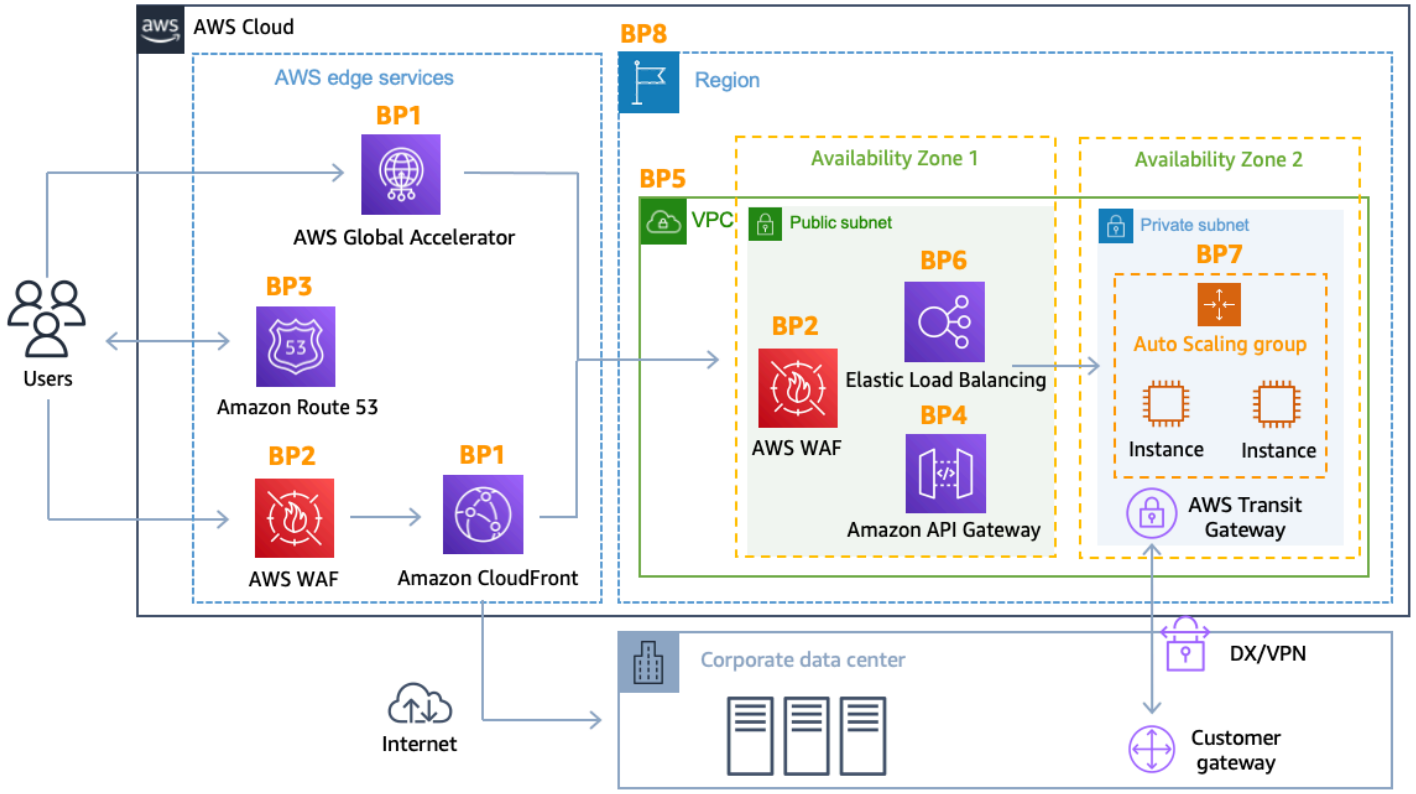
Some forms of DDoS mitigation are included automatically with AWS services. DDoS resilience can be improved further by using an AWS architecture with specific services, covered in the following sections, and by implementing additional best practices for each part of the network flow between users and your application.

You can use AWS services that operate from edge locations, such as Amazon CloudFront, AWS Global Accelerator, and Amazon Route 53 to build comprehensive availability protection against all known infrastructure layer attacks. These services are part of the AWS Global Edge Network, and can improve the DDoS resilience of your application when serving any type of application traffic from edge locations distributed around the world. You can run your application in any AWS Region, and use these services to protect your application availability and optimize the performance of your application for legitimate end users.

Benefits of using Amazon CloudFront, Global Accelerator, and Amazon Route 53 include:

- Access to internet and DDoS mitigation capacity across the AWS Global Edge Network. This is useful in mitigating larger volumetric attacks, which can reach terabit scale.
- AWS Shield DDoS mitigation systems are integrated with AWS edge services, reducing time-to-mitigate from minutes to sub second.
- Stateless SYN Flood mitigation verifies incoming connections using SYN cookies before passing them to the protected service. This ensures that only valid connections reach your application while protecting your legitimate end users against false positives drops.
- Automatic traffic engineering systems that disperse or isolate the impact of large volumetric DDoS attacks. All of these services isolate attacks at the source before they reach your origin, which means less impact on systems protected by these services.
- Application layer defense for CloudFront when combined with AWS WAF that does not require changing current application architecture (for example, in an AWS Region or on-premises data center).

There is no charge for inbound data transfer on AWS and you do not pay for DDoS attack traffic that is mitigated by AWS Shield. The following architecture diagram includes AWS Global Edge Network services.



DDoS-resilient reference architecture

This architecture includes several AWS services that can help you improve your web application's resiliency against DDoS attacks. The following table provides a summary of these services and the capabilities that they can provide. AWS has tagged each service with a best practice indicator (BP1, BP2) for easier reference within this document. For example, an upcoming section discusses the capabilities provided by Amazon CloudFront and Global Accelerator that includes the best practice indicator BP1.

Table 2 - Summary of best practices

	AWS Edge			AWS Region		
	Using Amazon CloudFront (BP1) with AWS WAF (BP2)	Using Global Accelerator (BP1)	Using Amazon Route 53 (BP3)	Using Elastic Load Balancing (BP6) with	Using security groups and network ACLs in	Using Amazon Elastic Compute Cloud (Amazon

	AWS Edge			AWS Region		
				AWS WAF (BP2)	Amazon VPC (BP5)	EC2) Auto Scaling (BP7)
Layer 3 (for example, UDP reflection) attack mitigation	✓	✓	✓	✓	✓	✓
Layer 4 (for example, SYN flood) attack mitigation	✓	✓	✓	✓		
Layer 6 (for example, TLS) attack mitigation	✓	✓	✓	✓		
Reduce attack surface	✓	✓	✓	✓	✓	
Scale to absorb application layer traffic	✓	✓	✓	✓	✓	✓

	AWS Edge			AWS Region		
Layer 7 (application layer) attack mitigation	✓	✓(*)	✓	✓	✓(*)	✓(*)
Geographic isolation and dispersion of excess traffic and larger DDoS attacks	✓	✓	✓			

✓(*): If used with AWS WAF with Application Load Balancer

Another way to improve your readiness to respond to and mitigate DDoS attacks is by subscribing to AWS Shield Advanced. Benefits of using AWS Shield Advanced include:

- Access to 24x7 specialized support from the AWS Shield Response Team (AWS SRT) for assistance with mitigating DDoS attacks that impact application availability, including an optional Proactive engagement feature
- Sensitive detection thresholds that route traffic into the DDoS mitigation system earlier and can improve time-to-mitigate attacks against Amazon EC2 (including elastic Load Balancer) or Network Load Balancer, when used with an Elastic IP address
- Tailored Layer 7 detection based on baselined traffic patterns of your application when used with AWS WAF
- Automatic application layer DDoS mitigation where Shield Advanced responds to detected DDoS attacks by creating, evaluating, and deploying custom AWS WAF rules
- Access to AWS WAF at no additional cost for the mitigation of application layer DDoS attacks (when used with Amazon CloudFront or Application Load Balancer)

- Centralized management of security policies through AWS Firewall Manager at no additional cost.
- Cost protection that enables you to request a limited refund of scaling-related costs that result from a DDoS attack.
- Enhanced service level agreement that is specific to AWS Shield Advanced customers.
- Protection groups that enable you to bundle resources, providing a self-service way to customize the scope of detection and mitigation for your application by treating multiple resources as a single unit. For information about protection groups, refer to Shield Advanced protection groups.
- DDoS attack visibility by using the AWS Management Console, API, and Amazon CloudWatch metrics and alarms.

This optional DDoS mitigation service helps protect applications hosted on any AWS Region. The service is available globally for CloudFront, Route 53, and Global Accelerator. Regionally, you can protect Application Load Balancer, Classic Load Balancer and Elastic IP addresses which allows you to protect Network Load Balancer (NLBs) or Amazon EC2 instances.

For a complete list of AWS Shield Advanced features and for more information about AWS Shield, refer to How AWS Shield works.

Best practices for DDoS mitigation

In the following sections, each of the recommended best practices for DDoS mitigation are described in more depth. For a quick and easy-to-implement guide on building a DDoS mitigation layer for static or dynamic web applications, refer to How to Help Protect Dynamic Web Applications Against DDoS Attacks by Using Amazon CloudFront and Amazon Route 53.

Infrastructure layer defense (BP1, BP3, BP6, BP7)

In a traditional datacenter environment, you can mitigate infrastructure layer DDoS attacks by using techniques like overprovisioning capacity, deploying DDoS mitigation systems, or scrubbing traffic with the help of DDoS mitigation services. On AWS, DDoS mitigation capabilities are automatically provided; but you can optimize your application's DDoS resilience by making architecture choices that best leverage those capabilities and also allow you to scale for excess traffic.

Key considerations to help mitigate volumetric DDoS attacks include ensuring that enough transit capacity and diversity are available and protecting AWS resources, like Amazon EC2 instances, against attack traffic.

Some Amazon EC2 instance types support features that can more easily handle large volumes of traffic, for example, up to 100 Gbps network bandwidth interfaces and enhanced networking. This helps prevent interface congestion for traffic that has reached the Amazon EC2 instance. Instances that support enhanced networking provide higher input/output (I/O) performance, higher bandwidth, and lower CPU utilization compared to traditional implementations. This improves the ability of the instance to handle large volumes of traffic and ultimately makes them highly resilient against packets per second (pps) load.

To allow this high level of resilience, AWS recommends using Amazon EC2 Dedicated Instances, or Amazon EC2 instances with higher networking throughput that have an "N" suffix and support for Enhanced Networking with up to 100 Gbps of Network bandwidth, for example, `c6gn.16xlarge` and `c5n.18xlarge` or metal instances (such as `c5n.metal`).

For more information about Amazon EC2 instances that support 100 Gigabit network interfaces and enhanced networking, refer to Amazon EC2 Instance Types.

The module required for enhanced networking and the required `enaSupport` attribute set are included with Amazon Linux 2 and the latest versions of the Amazon Linux AMI. Therefore, if you launch an instance with a hardware virtual machine (HVM) version of Amazon Linux on a supported instance type, enhanced networking is already enabled for your instance. For more information, refer to Test whether enhanced networking is enabled and Enhanced networking on Linux.

Amazon EC2 with Auto Scaling (BP7)

Another way to mitigate both infrastructure and application layer attacks is to operate at scale. If you have web applications, you can use load balancers to distribute traffic to a number of Amazon EC2 instances that are overprovisioned or configured to automatically scale. These instances can handle sudden traffic surges that occur for any reason, including a flash crowd or an application layer DDoS attack. You can set Amazon CloudWatch alarms to initiate Auto Scaling to automatically scale the size of your Amazon EC2 fleet in response to events that you define, such as CPU, RAM, Network I/O, and even custom metrics.

This approach protects application availability when there's an unexpected increase in request volume. When using Amazon CloudFront, Application Load Balancer, Classic Load Balancers, or Network Load Balancer with your application, TLS negotiation is handled by the distribution

(Amazon CloudFront) or by the load balancer. These features help protect your instances from being impacted by TLS-based attacks by scaling to handle legitimate requests and TLS abuse attacks.

For more information about using Amazon CloudWatch to invoke Auto Scaling, refer to Monitoring Amazon CloudWatch metrics for your Auto Scaling groups and instances.

Amazon EC2 provides resizable compute capacity so that you can quickly scale up or down as requirements change. You can scale horizontally by automatically adding instances to your application by scaling the size of your Amazon EC2 Auto Scaling group, and you can scale vertically by using larger EC2 instance types.

By using Amazon RDS Proxy, you can allow your applications to pool and share database connections to improve their ability to scale and handle unpredictable surges in database traffic. You can also enable storage auto-scaling for an Amazon RDS database instance. See Managing capacity automatically with Amazon RDS storage autoscaling for more information.

Elastic Load Balancing (BP6)

Large DDoS attacks can overwhelm the capacity of a single Amazon EC2 instance. With Elastic Load Balancing (ELB), you can reduce the risk of overloading your application by distributing traffic across many backend instances. Elastic Load Balancing can scale automatically, allowing you to manage larger volumes when you have unanticipated extra traffic, for example, due to flash crowds or DDoS attacks. For applications built within an Amazon VPC, there are three types of ELBs to consider, depending on your application type: Application Load Balancer (ALB), Network Load Balancer (NLB) and Classic Load Balancer (CLB).

For web applications, you can use the Application Load Balancer to route traffic based on content and accept only well-formed web requests. Application Load Balancer blocks many common DDoS attacks, such as SYN floods or UDP reflection attacks, protecting your application from the attack. Application Load Balancer automatically scales to absorb the additional traffic when these types of attacks are detected. Scaling activities due to infrastructure layer attacks are transparent for AWS customers and do not affect your bill.

For more information about protecting web applications with Application Load Balancer, refer to Getting Started with Application Load Balancers.

For non HTTP/HTTPS applications, you can use Network Load Balancer to route traffic to targets (for example, Amazon EC2 instances) at ultra-low latency. One key consideration with Network

Load Balancer is that any TCP SYN or UDP traffic that reaches the load balancer on a valid listener will be routed to your targets, not absorbed, however this does not apply for TLS-listeners which terminate the TCP connection. For Network Load Balancers with TCP listeners we recommend deploying Global Accelerator to protect against SYN flood.

You can use Shield Advanced to configure DDoS protection for Elastic IP addresses. When an Elastic IP address is assigned per Availability Zone to the Network Load Balancer, Shield Advanced will apply the relevant DDoS protections for the Network Load Balancer traffic.

For more information about protecting TCP and UDP applications with Network Load Balancer, refer to Getting started with Network Load Balancers.

Note

Depending on the security group configuration, it requires the resource using the security to group to use connection tracking to track information about traffic, this can affect the load balancer ability to process new connections, as the number of tracked connections is limited.

A security group configuration that contains an ingress rule accepting traffic from any IP address (for example, `0.0.0.0/0` or `::/0`) but do not have a corresponding rule to allow the response traffic, causes the security group to use connection tracking information to allow the response traffic to be sent. In an event of an DDoS attack, the maximum number of tracked connections can be exhausted. To improve the DDoS resilience of your public-facing Application Load Balancer or Classic Load Balancer, ensure that the security group associated with your load balancer is configured to not use connection tracking (untracked connections), so the flow of traffic is not to subject to connection tracking limits.

For this, configure your security group with a rule that allows the inbound rule to accept TCP flows from any IP address (`0.0.0.0/0` or `::/0`), and add a corresponding rule in the outbound direction allowing this resource to send the response traffic (allow outbound range for any IP address `0.0.0.0/0` or `::/0`) for all ports (0-65535), so the response traffic is allowed based on the security group rule, and not on tracking information. With this configuration, Classic and Application Load Balancer are not subject to exhaust connection tracking limits that may affect establishing new connections to its load balancer nodes, and allows it to scale based on the increase in traffic in the event of a DDoS attack. More information about untracked connections can be found at: Security group connection tracking: Untracked connections.

Avoiding security group connection tracking only helps in cases where the DDoS traffic originates from a source that is allowed by the security group—DDoS traffic from sources

that are not allowed in the security group do not affect connection tracking. Reconfiguring your security groups to avoid connection tracking is not required in these cases, for example, if your security group allow list consists of IP ranges with which you have a high degree of trust, such as a company corporate firewall or trusted VPN egress IPs or CDNs.

Use AWS Edge locations for scale (BP1, BP3)

Access to highly-scaled, diverse internet connections can significantly increase your ability to optimize latency and throughput to users, to absorb DDoS attacks, and to isolate faults while minimizing the impact on your application's availability. AWS edge locations provide an additional layer of network infrastructure that provides these benefits to any web application that uses Amazon CloudFront, Global Accelerator and Amazon Route 53. With these services, you can comprehensively protect on the edge your applications running from AWS Regions.

Web application delivery at the edge (BP1)

Amazon CloudFront is a service that can be used to deliver your entire website including static, dynamic, streaming, and interactive content. Persistent connections and variable time-to-live (TTL) settings can be used to offload traffic from your origin, even if you are not serving cacheable content. Use of these CloudFront features reduces the number of requests and TCP connections back to your origin, helping protect your web application from HTTP floods.

CloudFront only accepts well-formed connections, which helps prevent many common DDoS attacks, such as SYN floods and UDP reflection attacks, from reaching your origin. DDoS attacks are also geographically isolated close to the source, which prevents the traffic from impacting other locations. These capabilities can greatly improve your ability to continue serving traffic to users during large DDoS attacks. You can use CloudFront to protect an origin on AWS or elsewhere on the internet.

If you're using Amazon Simple Storage Service (Amazon S3) to serve static content on the internet, AWS recommends you use Amazon CloudFront to protect your bucket providing the following benefits:

- Restricts access to the Amazon S3 bucket so that it's not publicly accessible.
- Makes sure that viewers (users) can access the content in the bucket only through the specified CloudFront distribution—that is, prevents them from accessing the content directly from the bucket, or through an unintended CloudFront distribution.

To achieve this, configure CloudFront to send authenticated requests to Amazon S3, and configure Amazon S3 to only allow access to authenticated requests from CloudFront. CloudFront provides two ways to send authenticated requests to an Amazon S3 origin: origin access control (OAC) and origin access identity (OAI). We recommend using OAC because it supports:

- All Amazon S3 buckets in all AWS Regions, including opt-in Regions launched after December 2022
- Amazon S3 server-side encryption with AWS KMS (SSE-KMS)
- Dynamic requests (PUT and DELETE) to Amazon S3

For more information about OAC and OAI, refer to Restricting access to Amazon S3 origin.

For more information about protecting and optimizing the performance of web applications with Amazon CloudFront, refer to Getting Started with Amazon CloudFront.

Protect network traffic further from your origin using AWS Global Accelerator (BP1)

Global Accelerator is a networking service that improves availability and performance of users' traffic by up to 60%. This is accomplished by ingressing traffic at the edge location closest to your users and routing it over the AWS global network infrastructure to your application, whether it runs in a single or multiple AWS Regions.

Global Accelerator routes TCP and UDP traffic to the optimal endpoint based on performance in the closest AWS Region to the user. If there is an application failure, Global Accelerator provides failover to the next best endpoint within 30 seconds. Global Accelerator uses the vast capacity of the AWS global network and integrations with Shield, such as a stateless SYN proxy capability that challenges new connection attempts and only serves legitimate end users, to protect applications.

You can implement a DDoS resilient architecture that provides many of the same benefits as the Web Application Delivery at the Edge best practices, even if your application uses protocols not supported by CloudFront or you are operating a web application that requires global static IP addresses.

For example, you may require IP addresses that your end users can add to the allow list in their firewalls and are not used by any other AWS customers. In these scenarios you can use Global Accelerator to protect web applications running on Application Load Balancer and in conjunction with AWS WAF to also detect and mitigate web application layer request floods.

For more information about protecting and optimizing the performance of network traffic using Global Accelerator, refer to Getting started with Global Accelerator.

Domain name resolution at the edge (BP3)

Topics

Using Route 53 for DNS availability

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) service that can be used to direct traffic to your web application. It includes advanced features like Traffic Flow, Health Checks and Monitoring, Latency-Based Routing, and Geo DNS. These advanced features allow you to control how the service responds to DNS requests to improve the performance of your web application and to avoid site outages. It's the only AWS service that has a 100% data plane availability SLA.

Amazon Route 53 uses techniques such as shuffle sharding and anycast striping, that can help users access your application even if the DNS service is targeted by a DDoS attack.

With shuffle sharding, each name server in your delegation set corresponds to a unique set of edge locations and internet paths. This provides greater fault tolerance and minimizes overlap between customers. If one name server in the delegation set is unavailable, users can retry and receive a response from another name server at a different edge location.

Anycast striping allows each DNS request to be served by the most optimal location, dispersing the network load and reducing DNS latency. This provides a faster response for users. Additionally, Amazon Route 53 can detect anomalies in the source and volume of DNS queries, and prioritize requests from users that are known to be reliable.

For more information about using Amazon Route 53 to route users to your application, refer to Getting Started with Amazon Route 53.

Configuring Route 53 for cost protection from `NXDOMAIN` attacks

`NXDOMAIN` attacks occur when attackers send a flood of requests to a hosted zone for non-existent sub-domains, often via known "good" resolvers. The purpose of these attacks may be to impact

the cache of the recursive resolver and/or the availability of the authoritative resolver, or could be a form of DNS reconnaissance to try to discover hosted zone records. Using Route 53 for your authoritative resolver mitigates the risk of availability/performance impact, however the result can be a significant cost increase in monthly Route 53 costs. To protect against cost increases, take advantage of Route 53 pricing in which DNS queries are free when both of the following are true:

- The domain or subdomain name (`example.com` or `store.example.com`) and the record type (`A`) in the query match an alias record.
- The alias target is an AWS resource other than another Route 53 record.

Create a wildcard record, for example, `*.example.com` with a type `A` (Alias) pointing at an AWS resource such as an EC2 instance, Elastic Load Balancer or CloudFront distribution, so that when a query for `qwerty12345.example.com` is made, the IP of the resource will be returned and you will not be charged for the query.

Application layer defense (BP1, BP2)

Many of the techniques discussed so far in this paper are effective at mitigating the impact that infrastructure layer DDoS attacks have on your application's availability. To also defend against application layer attacks, you need to implement an architecture that allows you to specifically detect, scale to absorb, and block malicious requests. This is an important consideration because network-based DDoS mitigation systems are generally ineffective at mitigating complex application layer attacks.

Detect and filter malicious web requests (BP1, BP2)

When your application runs on AWS, you can leverage Amazon CloudFront (and its HTTP caching capability), AWS WAF, and Shield Advanced Automatic Application layer protection to help prevent unnecessary requests reach your origin during application layer DDoS attacks.

Amazon CloudFront

Amazon CloudFront can help reduce server load by preventing non-web traffic from reaching your origin. To send a request to a CloudFront application, the connection must be established with a valid IP address through a completed TCP handshake, which cannot be faked. Additionally, CloudFront can automatically close connections from slow reading or slow writing attackers (for example, Slowloris).

CDN caching

CloudFront allows you to serve both dynamic content and static content from AWS edge locations. By serving proxy cacheable content from CDN cache you prevent requests from reaching your origin from a given edge cache node for the duration of the caching TTL. In conjunction with request collapsing for expired but cacheable content, even very short TTL mean that negligible numbers of requests will reach your origin during request floods for that content. In addition enabling features like CloudFront Origin Shield can further help reduce the load on your origin – anything you can do to improve your cache hit ratio can mean the difference between an impactful and non-impactful request flood attack.

AWS WAF

By using AWS WAF, you can configure web access control lists (Web ACLs) on your global CloudFront distributions or regional resources to filter, monitor and block requests based on request signatures. To determine whether to allow or block requests, you can consider factors such as the IP address or country of origin, certain strings or patterns in the request, the size of specific parts of the request, and the presence of malicious SQL code or scripting. You can also run CAPTCHA puzzles and silent client session challenges against requests.

Both AWS WAF and CloudFront also enable you to set geo-restrictions to block or allow requests from selected countries. This can help block or rate-limit attacks from geographic locations where you do not expect to serve users. With fine-grained geographic match rule statements in AWS WAF, you can control access down to the region level.

You can use Scope-down statements to narrow the scope of the requests that the rule evaluates to save costs and "labels" on web requests to allow a rule that matches the request to communicate the match results to rules that are evaluated later in the same web ACL. Choose this option to reuse the same logic across multiple rules.

You can also define a complete custom response, with response code, headers, and body.

To help identify malicious requests, review your web server logs or use AWS WAF's logging and request sampling.By enabling AWS WAF logging, you get detailed information about the traffic analyzed by the Web ACL. AWS WAF supports log filtering, allowing you to specify which web requests are logged and which requests are discarded from the log after the inspection.

Information recorded in the logs includes the time that AWS WAF received the request from your AWS resource, detailed information about the request, and the matching action for each rule requested.

Sampled requests provide details about requests within the past three hours that matched one of your AWS WAF rules. You can use this information to identify potentially malicious traffic signatures and create a new rule to deny those requests. If you see a number of requests with a random query string, make sure to allow only the query string parameters that are relevant to cache for your application. This technique is helpful in mitigating a cache busting attack against your origin.

AWS WAF – Rate-based rules

AWS strongly recommends protecting against HTTP request floods by using the rate-based rules in AWS WAF to automatically block IP addresses of bad actors when the number of requests received in a 5-minute sliding window exceed a threshold that you define. Offending client IP addresses will receive a 403 forbidden response (or configured block error response) and remain blocked until request rates drop below the threshold.

It's recommended to layer rate-based rules to provide enhanced protection so that you have:

- A blanket rate-based rule to protect your application from large HTTP floods.
- One or more rate-based rules to protect specific URIs at more restrictive rates than the blanket rate-based rule.

For instance you may choose a blanket rate-based rule (no scope-down statement) with a limit of 500 requests within a 5-minute period, and then create one or more of the following rate-based rules with lower limits than 500 (as low as 100 requests in a 5-minute period) using scope-down statements:

- Protect your **web pages** with a scope-down statement like "`if NOT uri_path contains '.'`" so that requests for resources without a file extension are further protected. This also protects your homepage (*/*) which is a frequently targeted URI path.
- Protect **dynamic endpoints** with a scope-down statement like "`if method exactly matches 'post' (convert lowercase)`"
- Protect **heavy requests** that reach your database or invoke a one-time password (OTP) with a scope-down like "`if uri_path starts_with '/login' OR uri_path starts_with '/signup' OR uri_path starts_with '/forgotpassword'`"

Rate-based in "Block" mode are the cornerstone of your defense-in-depth WAF configuration to protect against request floods and are a requirement for AWS Shield Advanced cost protection

requests to be approved. We'll examine additional defense-in-depth WAF configurations in the following sections.

AWS WAF – IP reputation

To prevent attacks based on IP address reputation, you can create rules using IP matching or use Managed Rules for AWS WAF.

Amazon's IP reputation list rule group includes rules based on Amazon's internal threat intelligence. These rules look for IP addresses that are bots, performing reconnaissance against AWS resources, or actively engaging in DDoS activities. The `AWSManagedIPDDoSList` rule, has been observed blocking over 90% of malicious request floods.

The Anonymous IP list rule group contains rules to block requests from services that allow the obfuscation of viewer identity. These include requests from VPNs, proxies, Tor nodes, and cloud platforms (excluding AWS).

In addition you can make use of third-party IP reputation lists by using the IP Lists parser component of the Security Automations for AWS WAF solution.

AWS WAF - Intelligent threat mitigation

Botnets are a serious security threat and are commonly used to carry out illegal or harmful activities such as sending spam, stealing sensitive data, initiating ransomware attacks, committing ad fraud through fraudulent clicks, or launching distributed denial-of-service (DDoS) attacks. To prevent bot attacks, use the AWS WAF Bot Control managed rule group. This rule group provides a basic, "Common" protection level that adds labels to self-identifying bots, verifies generally desirable bots, and detects high confidence bot signatures and a "Targeted" protection level that adds detection for advanced bots that don't self-identify.

Targeted protections use advanced detection techniques such as browser interrogation, fingerprinting, and behavior heuristics to identify bad bot traffic and then applies mitigation controls such as rate limiting and CAPTCHA and Challenge rule actions. Targeted also provides rate limiting options to enforce human-like access patterns and apply dynamic rate limiting through the use of request tokens. For additional details, see AWS WAF Bot Control rule group. To detect and manage malicious takeover attempts on your application's login page, you can use AWS WAF Fraud Control account takeover prevention (ATP) rule group. The rule group does this by inspecting login attempts that clients send to your application's login endpoint and also inspects your application's responses to login attempts, to track success and failure rate.

Account creation fraud is an online illegal activity in which an attacker tries to create one or more fake accounts. Attackers use fake accounts for fraudulent activities such as abusing promotional and sign up bonuses, impersonating someone, and cyberattacks like phishing. The presence of fake accounts can negatively impact your business by damaging your reputation with customers and exposure to financial fraud.

You can monitor and control account creation fraud attempts by implementing the AWS WAF Fraud Control account creation fraud prevention (ACFP) feature. AWS WAF offers this feature in the AWS Managed Rules rule group `AWSManagedRulesACFPRuleSet` with companion application integration SDKs.

Learn more about these protections in AWS WAF intelligent threat mitigation.

Automatically mitigate application-layer DDoS events (BP1, BP2, BP6)

If you are subscribed to AWS Shield Advanced, you can enable Shield Advanced automatic application layer DDoS mitigation. This feature automatically creates, evaluates, and deploys AWS WAF rules to mitigate layer 7 DDoS events on your behalf.

AWS Shield Advanced establishes a traffic baseline for each protected resource associated with a WAF WebACL. Traffic that significantly deviates from the established baseline is flagged as a potential DDoS event. After an event is detected, AWS Shield Advanced attempts to identify a signature of the web requests that constitute the event, and if a signature is identified, AWS WAF rules are created to mitigate traffic with that signature.

Once rules are evaluated against the historical baseline and deemed to be safe, they are added to the Shield-managed rule group, and you can choose whether the rules are deployed in count or block mode. Shield Advanced automatically removes AWS WAF rules after it has determined that an event has fully subsided.

Engage SRT (Shield Advanced subscribers only)

In addition, when subscribed to Shield Advanced, you can engage the AWS SRT to help you create rules to mitigate an attack that is hurting your application's availability. You can grant AWS SRT limited access to your account's AWS Shield Advanced and AWS WAF APIs. AWS SRT accesses these APIs to place mitigations on your account only with your explicit authorization. For more information, refer to the Support section of this document.

You can use AWS Firewall Manager to centrally configure and manage security rules, such as AWS Shield Advanced protections and AWS WAF rules, across your organization. Your AWS

Organizations management account can designate an administrator account, which is authorized to create Firewall Manager policies. These policies allow you to define criteria, such as resource type and tags, which determine where rules are applied. This is useful when you have multiple accounts and want to standardize your protection.

For more information about:

- AWS Managed Rules for AWS WAF, refer to AWS Managed Rules for AWS WAF.
- Using geographic restriction to limit access to your CloudFront distribution, refer to Restricting the geographic distribution of your content.
- Using AWS WAF, refer to:
  - Getting started with AWS WAF
  - Logging web ACL traffic information
  - Viewing a sample of web requests
- Configuring rate-based rules, refer to Protect Web Sites and Services Using Rate-Based Rules for AWS WAF.
- How to manage the deployment of rules across your AWS resources with Firewall Manager, see:
  - Getting started with Firewall Manager AWS WAF policies.
  - Getting started with Firewall Manager Shield Advanced policies.

Attack surface reduction

Another important consideration when architecting an AWS solution is to limit the opportunities an attacker has to target your application. This concept is known as attack *surface reduction*. Resources that are not exposed to the internet are more difficult to attack, which limits the options an attacker has to target your application's availability.

For example, if you do not expect users to directly interact with certain resources, make sure that those resources are not accessible from the internet. Similarly, do not accept traffic from users or external applications on ports or protocols that aren't necessary for communication.

In the following section, AWS provides best practices to guide you in reducing your attack surface and limiting your application's internet exposure.

Obfuscating AWS resources (BP1, BP4, BP5)

Typically, users can quickly and easily use an application without requiring that AWS resources be fully exposed to the internet.

Security groups and network ACLs (BP5)

Amazon Virtual Private Cloud (Amazon VPC) allows you to provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define.

Security groups and network ACLs are similar in that they allow you to control access to AWS resources within your VPC. But security groups allow you to control inbound and outbound traffic at the instance level, while network ACLs offer similar capabilities at the VPC subnet level. There is no additional charge for using security groups or network ACLs.

You can choose whether to specify security groups when you launch an instance or associate the instance with a security group at a later time. All internet traffic to a security group is implicitly denied unless you create an *allow* rule to permit the traffic.

For example, when you have Amazon EC2 instances behind an Elastic Load Balancer, the instances themselves should not need to be publicly accessible and should have private IPs only. Instead, you could provide the Elastic Load Balancer access to the required target listener ports using a Security Group rule that allows access to 0.0.0.0/0 (to avoid connection tracking issues – see note below) in conjunction with a Network Access Control List (NACL) on the target group subnet to allow only

the Elastic Load Balancing IP ranges to communicate with the instances. This ensures that internet traffic can't directly communicate with your Amazon EC2 instances, which makes it more difficult for an attacker to learn about and impact your application.

When you create network ACLs, you can specify both allow and deny rules. This is useful if you want to explicitly deny certain types of traffic to your application. For example, you can define IP addresses (as CIDR ranges), protocols, and destination ports that are denied access to the entire subnet. If your application is used only for TCP traffic, you can create a rule to deny all UDP traffic, or vice versa. This option is useful when responding to DDoS attacks because it lets you create your own rules to mitigate the attack when you know the source IPs or other signature.

If you are subscribed to AWS Shield Advanced, you can register Elastic IP addresses as protected resources. DDoS attacks against Elastic IP addresses that have been registered as protected resources are detected more quickly, which can result in a faster time to mitigate. When an attack is detected, the DDoS mitigation systems reads the network ACL that corresponds to the targeted Elastic IP address and enforces it at the AWS network border, rather than at the subnet level. This significantly reduces your risk of impact from a number of infrastructure layer DDoS attacks.

For more information about configuring security groups and network ACLs to optimize for DDoS resiliency, refer to How to Help Prepare for DDoS Attacks by Reducing Your Attack Surface.

For more information about using Shield Advanced with Elastic IP addresses as protected resources, refer to the steps to Subscribe to AWS Shield Advanced.

Protecting your origin (BP1, BP5)

If you are using Amazon CloudFront with an origin that is inside of your VPC, you may want to ensure that only your CloudFront distribution can forward requests to your origin. With Edge-to-Origin Request Headers, you can add or override the value of existing request headers when CloudFront forwards requests to your origin. You can use the *Origin Custom Headers*, for example, the `X-Shared-Secret` header, to help validate that the requests made to your origin were sent from CloudFront.

For more information about protecting your origin with an *Origin Custom Headers*, refer to Adding custom headers to origin requests and Restricting access to Application Load Balancers.

For a guide on implementing a sample solution to automatically rotate the value of *Origin Custom Headers* for the origin access restriction, refer to How to enhance Amazon CloudFront origin security with AWS WAF and Secrets Manager.

Alternatively, you can use an AWS Lambda function to automatically update your security group rules to allow only CloudFront traffic. This improves your origin's security by helping to ensure that malicious users cannot bypass CloudFront and AWS WAF when accessing your web application.

For more information about how to protect your origin by automatically updating your security groups, and the `X-Shared-Secret` header, refer to How to Automatically Update Your Security Groups for Amazon CloudFront and AWS WAF by Using AWS Lambda.

However, the solution involves additional configuration and the cost of running Lambda functions. To simplify this, we have now introduced an AWS-managed prefix list for CloudFront to limit the inbound HTTP/HTTPS traffic to your origins from only the CloudFront origin-facing IP addresses. AWS-managed prefix lists are created and maintained by AWS and are available to use at no additional cost. You can reference the managed prefix list for CloudFront in your (Amazon VPC) security group rules, subnet route tables, common security group rules with AWS Firewall Manager, and any other AWS resources that can use a managed prefix list.

For more information about using AWS-managed prefix list for Amazon CloudFront, refer to Limit access to your origins using the AWS-managed prefix list for Amazon CloudFront.

> **Note**
>
> As discussed in other sections of this document, relying on security groups to protect your origin can add security-group connection tracking as a potential bottle-neck during a request flood. Unless you are able to filter malicious requests at CloudFront using a caching policy that enables caching, it may be better to rely on the *Origin Custom Headers*, discussed previously, to help validate that the requests made to your origin were sent from CloudFront, rather than use security groups. Using a custom request header with an Application Load Balancer listener rule prevents throttling due to tracking limits that may affect establishing new connections to a load balancer, thus allowing Application Load Balancer to scale based on the increase in traffic in an event of an DDoS attack.

Protecting API endpoints (BP4)

When you must expose an API to the public, there is a risk that the API frontend could be targeted by a DDoS attack. To help reduce the risk, you can use Amazon API Gateway as an entryway to applications running on Amazon EC2, AWS Lambda, or elsewhere. By using Amazon API Gateway, you don't need your own servers for the API frontend and you can obfuscate other components

of your application. By making it harder to detect your application's components, you can help prevent those AWS resources from being targeted by a DDoS attack.

When you use Amazon API Gateway, you can choose from two types of API endpoints. The first is the default option: edge-optimized API endpoints that are accessed through an Amazon CloudFront distribution. The distribution is created and managed by API Gateway, however, so you don't have control over it. The second option is to use a regional API endpoint that is accessed from the same AWS Region in which your REST API is deployed. AWS recommends that you use the second type of endpoint and associate it with your own Amazon CloudFront distribution. This gives you control over the Amazon CloudFront distribution and the ability to use AWS WAF for application layer protection. This mode provides you with access to scaled DDoS mitigation capacity across the AWS global edge network.

When using Amazon CloudFront and AWS WAF with Amazon API Gateway, configure the following options:

- Configure the cache behavior for your distributions to forward all headers to the API Gateway regional endpoint. By doing this, CloudFront will treat the content as dynamic and skip caching the content.
- Protect your API Gateway against direct access by configuring the distribution to include the origin custom header x-api-key, by setting the API key value in API Gateway.
- Protect the backend from excess traffic by configuring standard or burst rate limits for each method in your REST APIs.

For more information about creating APIs with Amazon API Gateway, refer to Amazon API Gateway Getting Started.

Operational techniques

The mitigation techniques in this paper help you architect applications that are inherently resilient against DDoS attacks. In many cases, it's also useful to know when a DDoS attack is targeting your application so you can take mitigation steps. This section discusses best practices for gaining visibility into abnormal behavior, alerting and automation, managing protection at scale, and engaging AWS for additional support.

Load testing

Regularly load test your application using the guidelines in our Load Testing Applications whitepaper with both expected and above expected traffic levels so you can see how effective your architecture is, how your Auto Scaling policies function and how your error handling functions. Test for expected traffic scale-up and down but also for "flash-crowd" type behavior. Retest either periodically or before any major release. For layer 3 or 4 DDoS simulation testing, such as SYN flood, follow our DDoS Simulation Testing Policy.

Metrics and alarms

As a best practice you should be using infrastructure and application monitoring tools to check the availability of your application to ensure your application is not impacted by a DDoS event, as an option you can configure application and infrastructure Route 53 health checks for the resources to help improve the detection of DDoS events. For more information about health checks, see AWS WAF, Firewall Manager and Shield Advanced Developer Guide.

When a key operational metric deviates substantially from the expected value, an attacker may be attempting to target your application's availability. Familiarity with the normal behavior of your application, means you can take action more quickly when you detect an anomaly. Amazon CloudWatch can help by monitoring applications that you run on AWS. For example, you can collect and track metrics, collect and monitor log files, set alarms, and automatically respond to changes in your AWS resources.

If you follow the DDoS-resilient reference architecture when architecting your application, common infrastructure layer attacks will be blocked before reaching your application. If you are subscribed to AWS Shield Advanced, you have access to a number of CloudWatch metrics that can indicate that your application is being targeted.

For example, you can configure alarms to notify you when there is a DDoS attack in progress, so you can check your application's health and decide whether to engage AWS SRT. You can configure the `DDoSDetected` metric to tell you if an attack has been detected. If you want to be alerted based on the attack volume, you can also use the `DDoSAttackBitsPerSecond`, `DDoSAttackPacketsPerSecond`, or `DDoSAttackRequestsPerSecond` metrics. You can monitor these metrics by integrating CloudWatch with your own tools or by using tools provided by third parties, such as Slack or PagerDuty.

An application layer attack can elevate many Amazon CloudWatch metrics. If you're using AWS WAF, you can use CloudWatch to monitor and activate alarms on increases in requests that you've set in AWS WAF to be allowed, counted, or blocked. This allows you to receive a notification if the level of traffic exceeds what your application can handle. You can also use Amazon CloudFront, Amazon Route 53, Application Load Balancer, Network Load Balancer, Amazon EC2, and Auto Scaling metrics that are tracked in CloudWatch to detect changes that can indicate a DDoS attack.

The following table lists descriptions of CloudWatch metrics that are commonly used to detect and react to DDoS attacks.

Table 3 - Recommended Amazon CloudWatch metrics

Topic	Metric	Description
AWS Shield Advanced	`DDoSDetected`	Indicates a DDoS event for a specific Amazon Resource Name (ARN).
AWS Shield Advanced	`DDoSAttackBitsPerSecond`	The number of bytes observed during a DDoS event for a specific ARN. This metric is only available for layer 3 or 4 DDoS events.
AWS Shield Advanced	`DDoSAttackPacketsPerSecond`	The number of packets observed during a DDoS event for a specific ARN. This metric is only available for layer 3 or 4 DDoS events.

Topic	Metric	Description
AWS Shield Advanced	`DDoSAttackRequests PerSecond`	The number of requests observed during a DDoS event for a specific ARN. This metric is only available for layer 7 DDoS events and is only reported for the most significant layer 7 events.
AWS WAF	`AllowedRequests`	The number of allowed web requests.
AWS WAF	`BlockedRequests`	The number of blocked web requests.
AWS WAF	`CountedRequests`	The number of counted web requests.
AWS WAF	`PassedRequests`	The number of passed requests. This is only used for requests that go through a rule group evaluation without matching any of the rule group rules.
Amazon CloudFront	`Requests`	The number of HTTP/S requests.
Amazon CloudFront	`TotalErrorRate`	The percentage of all requests for which the HTTP status code is `4xx` or `5xx`.
Amazon Route 53	`HealthCheckStatus`	The status of the health check endpoint.

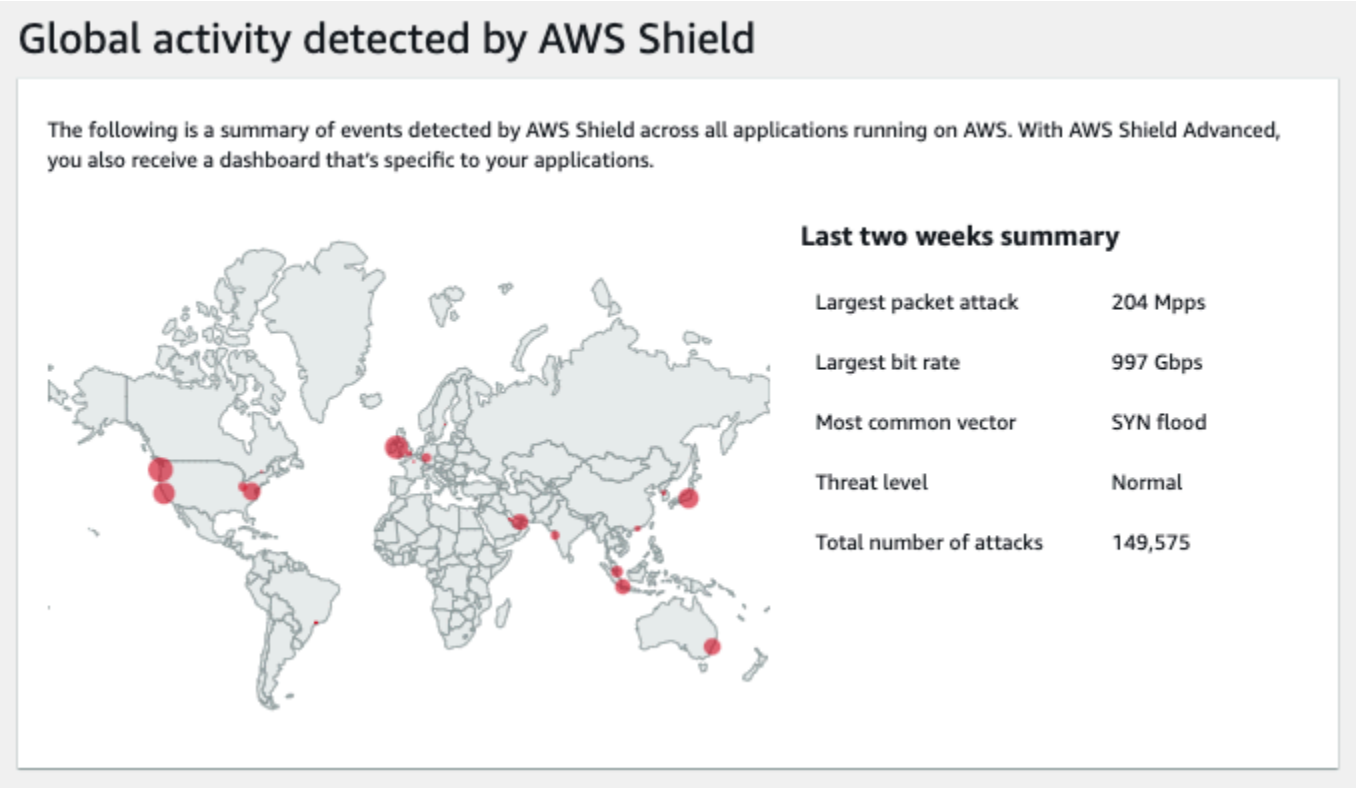
Topic	Metric	Description
Application Load Balancer	ActiveConnectionCount	The total number of concurrent TCP connections that are active from clients to the load balancer, and from the load balancer to targets.
Application Load Balancer	ConsumedLCUs	The number of load balancer capacity units (LCU) used by your load balancer.
Application Load Balancer	HTTPCode_ELB_4XX_Count HTTPCode_ELB_5XX_Count	The number of HTTP 4xx or 5xx client error codes generated by the load balancer.
Application Load Balancer	NewConnectionCount	The total number of new TCP connections established from clients to the load balancer, and from the load balancer to targets.
Application Load Balancer	ProcessedBytes	The total number of bytes processed by the load balancer.
Application Load Balancer	RejectedConnectionCount	The number of connections that were rejected because the load balancer had reached its maximum number of connections.
Application Load Balancer	RequestCount	The number of requests that were processed.

Topic	Metric	Description
Application Load Balancer	`TargetConnectionErrorCount`	The number of connections that were not successfully established between the load balancer and the target.
Application Load Balancer	`TargetResponseTime`	The time elapsed, in seconds, after the request left the load balancer until a response from the target is received.
Application Load Balancer	`UnHealthyHostCount`	The number of targets that are considered unhealthy.
Network Load Balancer	`ActiveFlowCount`	The total number of concurrent TCP flows (or connections) from clients to targets.
Network Load Balancer	`ConsumedLCUs`	The number of load balancer capacity units (LCU) used by your load balancer.
Network Load Balancer	`NewFlowCount`	The total number of new TCP flows (or connections) established from clients to targets in the time period.
Network Load Balancer	`ProcessedBytes`	The total number of bytes processed by the load balancer, including TCP/IP headers.
Global Accelerator	`NewFlowCount`	The total number of new TCP and UDP flows (or connections) established from clients to endpoints in the time period.

Topic	Metric	Description
Global Accelerator	`ProcessedBytesIn`	The total number of incoming bytes processed by the accelerator, including TCP/IP headers.
Auto Scaling	`GroupMaxSize`	The maximum size of the Auto Scaling group.
Amazon EC2	`CPUUtilization`	The percentage of allocated EC2 compute units that are currently in use.
Amazon EC2	`NetworkIn`	The number of bytes received by the instance on all network interfaces.

For more information about using Amazon CloudWatch to detect DDoS attacks on your application, refer to Getting Started with Amazon CloudWatch.

AWS includes several additional metrics and alarms to notify you about an attack and to help you monitor your application's resources. The AWS Shield console or API provide a per-account event summary and details about attacks that have been detected.



Global activity detected by AWS Shield

In addition, the global threat environment dashboard provides summary information about all DDoS attacks that have been detected by AWS. This information may be useful to better understand DDoS threats across a larger population of applications in addition to attack trends, and comparing with attacks that you may have observed.

If you are subscribed to AWS Shield Advanced, the service dashboard displays additional detection and mitigation metrics and network traffic details for events detected on protected resources. AWS Shield evaluates traffic to your protected resource along multiple dimensions. When an anomaly is detected, AWS Shield creates an event and reports the traffic dimension where the anomaly was observed. With a placed mitigation this protects your resource from receiving excess traffic and traffic that matches a known DDoS event signature.

Detection metrics are based on sampled network flows or AWS WAF logs when a web ACL is associated with the protected resource. Mitigation metrics are based on traffic that's observed by Shield's DDoS mitigation systems. Mitigation metrics are a more precise measurement of the traffic into your resource.

The *network top contributors* metric provides insight into where traffic is coming from during a detected event. You can view the highest volume contributors and sort by aspects such as protocol,

source port, and TCP flags. The top contributors metric includes metrics for all traffic observed on the resource along various dimensions. It provides additional metric dimensions you can use to understand network traffic that's sent to your resource during an event. Keep in mind that for non-reflection layer 3 or 4 attacks, the source IP addresses may have been spoofed and cannot be relied on.

The service dashboard also includes details about the actions automatically taken to mitigate DDoS attacks. This information makes it easier to investigate anomalies, explore dimensions of the traffic, and better understand the actions taken by Shield Advanced to protect your availability.

Logging

Enable useful logging on all services according to our Logging and monitoring guide for application owners to maximize visibility and assist with troubleshooting. This includes, but is not limited to:

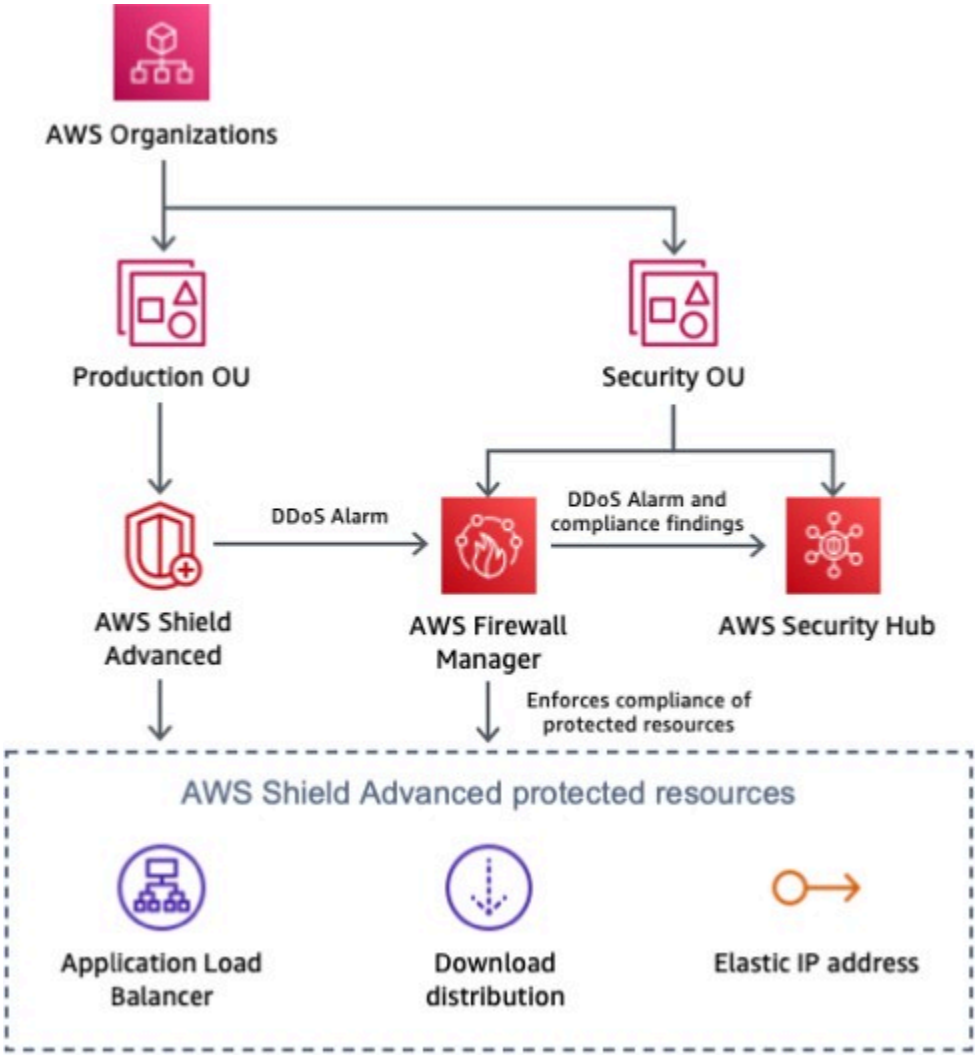
- AWS CloudTrail
- AWS WAF Logs
- CloudFront access logs
- VPC Flow Logs (see Log and View Network Traffic Flows) – include `tcp-flags` field in the included fields to maximize visibility
- ELB access logs (ALB, CLB, NLB)
- Web server HTTP access logs
- Operating system security logging
- Application logging

Visibility and protection management across multiple accounts

In scenarios when you operate across multiple AWS accounts and have multiple components to protect, using techniques that enable you to operate at scale and reduce operational overhead increase your mitigation capabilities. When managing AWS Shield Advanced protected resources in multiple accounts, you can set up centralized monitoring by using AWS Firewall Manager and AWS Security Hub. With Firewall Manager, you can create a security policy that enforces DDoS protection compliance across all your accounts. You can use these two services together to manage your protected resources across multiple accounts and centralize the monitoring of those resources.

Security Hub automatically integrates with Firewall Manager, allowing Shield Advanced customers to view security findings in a single dashboard, alongside other high priority security alerts and compliance statuses.

For example, when Shield Advanced detects anomalous traffic destined for a protected resource in any AWS account within the scope, this finding will be visible in the Security Hub console. If configured, Firewall Manager can automatically bring the resource into compliance by creating it as a Shield Advanced–protected resource, and then update Security Hub when the resource is in a compliant state.



Architecture diagram showing monitoring AWS Shield-protected resources with Firewall Manager and Security Hub

For more information about central monitoring of Shield protected resources, refer to Set up centralized monitoring for DDoS events and auto-remediate noncompliant resources.

Incident response strategy and runbooks

Developing a DDoS attack incident response strategy and building a security incident response process around it is crucial for all organizations. A recommended approach is to model your response playbook based on NIST's suggested steps such as gathering evidence, mitigating, recovering, and conducting post-incident analysis. For example, a response playbook for web application DoS or DDoS attacks is provided as an example. Additional resources are available in the AWS Security Incident Response Guide.

Support

If you experience an attack, you can also benefit from support from AWS in assessing the threat and reviewing the architecture of your application, or you might want to request other assistance. It is important to create a response plan for DDoS attacks before an actual event. The best practices outlined in this paper are intended to be proactive measures that you implement before you launch an application, but DDoS attacks against your application might still occur. Review the options in this section to determine the support resources that are best suited for your scenario. Your account team can evaluate your use case and application, and assist with specific questions or challenges that you have.

If you're running production workloads on AWS, consider subscribing to Business Support, which provides you with 24/7 access to Cloud Support Engineers who can assist with DDoS attack issues. If you're running mission critical workloads, consider Enterprise Support which provides the ability to open critical cases and receive the fastest response from a Senior Cloud Support Engineer.

If you're subscribed to AWS Shield Advanced and are also subscribed to either Business Support or Enterprise Support, you can configure Shield proactive engagement. It allows you to configure health checks, associate to your resources, and provide 24/7 operations contact information. When Shield detects signs of DDoS and your application health checks are showing signs of degradation, AWS SRT will proactively reach out to you. This is our recommended engagement model because it allows for the quickest AWS SRT response times and empowers AWS SRT to begin troubleshooting even before contact has been established with you.

For more information, refer to Compare AWS Support Plans.

The proactive engagement feature requires you to configure a Route 53 health check that accurately measures the health of your application and is associated with the resource protected by Shield Advanced. Once a Route 53 health check is associated in the Shield console, the Shield

Advanced detection system uses the health check status as an indicator of your application's health. The health-based detection feature in Shield Advanced will ensure that you are notified and that mitigations are placed more quickly when your application is unhealthy. AWS SRT will contact you to troubleshoot whether the unhealthy application is being targeted by a DDoS attack and place additional mitigations as needed.

Completing configuration of proactive engagement includes adding contact details in the Shield console. AWS SRT will use this information to contact you. You can configure up to ten contacts, and provide additional notes if you have any specific contact requirements or preferences. Proactive

engagement contacts should hold a 24/7 role, such as a security operations center or an individual who is immediately available.

You can enable proactive engagement for all resources or for select key production resources where response time is critical. This is accomplished by assigning health checks only to these resources.

You can also escalate to AWS SRT by creating an AWS Support case using the AWS Support console (sign-in required), or Support API if you have a DDoS-related event that affects your application's availability.

Conclusion

The best practices outlined in this paper can help you build a DDoS resilient architecture that protects your application's availability by preventing many common infrastructure and application layer DDoS attacks. The extent to which you follow these best practices when you architect your application will influence the type, vector, and volume of DDoS attacks that you can mitigate. You can incorporate resiliency without subscribing to a DDoS mitigation service. By choosing to subscribe to AWS Shield Advanced you gain additional support, visibility, mitigation, and cost protection features that further protect an already resilient application architecture.

Contributors

Contributors to this document include:

- Rodrigo Ferroni, AWS Security Specialist TAM
- Dmitriy Novikov, AWS Solutions Architect
- Achraf Souk, AWS Solutions Architect
- Joanna Knox, AWS Support Engineering
- Anuj Butail, AWS Solution Architect
- Harith Gaddamanugu, AWS Edge Specialist SA

Further reading

For additional information, refer to:

- Guidelines for Implementing AWS WAF (AWS Whitepaper)
- NIS301 – re:Inforce 2023: How AWS threat intelligence becomes managed firewall rules (YouTube video)
- NET314 - re:Invent 2022: Building DDoS-resilient applications using AWS Shield (YouTube video)
- SEC321 - re:Invent 2020: Get ahead of the curve with DDoS Response Team escalations (YouTube video)
- William Hill: High-performance DDoS Protection with AWS - 2020 (YouTube video)
- SEC407 - re:Invent 2019: A defense-in-depth approach to building web applications (YouTube video)
- Best Practices for DDoS Mitigation on AWS – 2018 (YouTube video)
- SID324 – re:Invent 2017: Automating DDoS Response in the Cloud (YouTube video)
- CTD304 – re:Invent 2017: Dow Jones & Wall Street Journal's Journey to Manage Traffic Spikes While (YouTube video)
- Mitigating DDoS & Application Layer Threats (YouTube video)
- CTD310 – re:Invent 2017: Living on the Edge, It's Safer Than You Think! Building Strong with Amazon (YouTube video)
- CloudFront, AWS Shield, and AWS WAF (YouTube video)

Document revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Whitepaper update	Added OAC for CloudFront and DNS wildcard cost protection. Expanded discussion of operational techniques, caching, rate-based rules and managed rule groups. Added on-premises into architecture diagram, removed duplication, and clarified text to remove ambiguity.	August 9, 2023
Whitepaper update	Revised for clarity; Updated to include latest recommendations and features: Security group connection tracking and Shield Advanced automatic application layer DDoS mitigation.	April 13, 2022
Whitepaper update	Updated to include latest recommendations and features. AWS Global Accelerator is added as part of comprehensive protection at the edge. AWS Firewall Manager for centralized monitoring for DDoS events and auto-remediate non-compliant resources.	September 21, 2021

Whitepaper update	Updated to clarify cache busting in *Detect and Filter Malicious Web Requests (BP1, BP2)* section, and ELB and ALB usage in *Scale to Absorb (BP6)* section. Updated diagrams and Table 2, marked "Choice of Region." as BP8. Updated BP7 section with more details.	December 18, 2019
Whitepaper update	Updated to include AWS WAF logging as a best practice.	December 1, 2018
Whitepaper update	Updated to include AWS Shield, AWS WAF features, AWS Firewall Manager, and related best practices.	June 1, 2018
Whitepaper update	Added prescriptive architect ure guidance and updated to include AWS WAF.	June 1, 2016
Initial publication	Whitepaper published.	June 1, 2015

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.