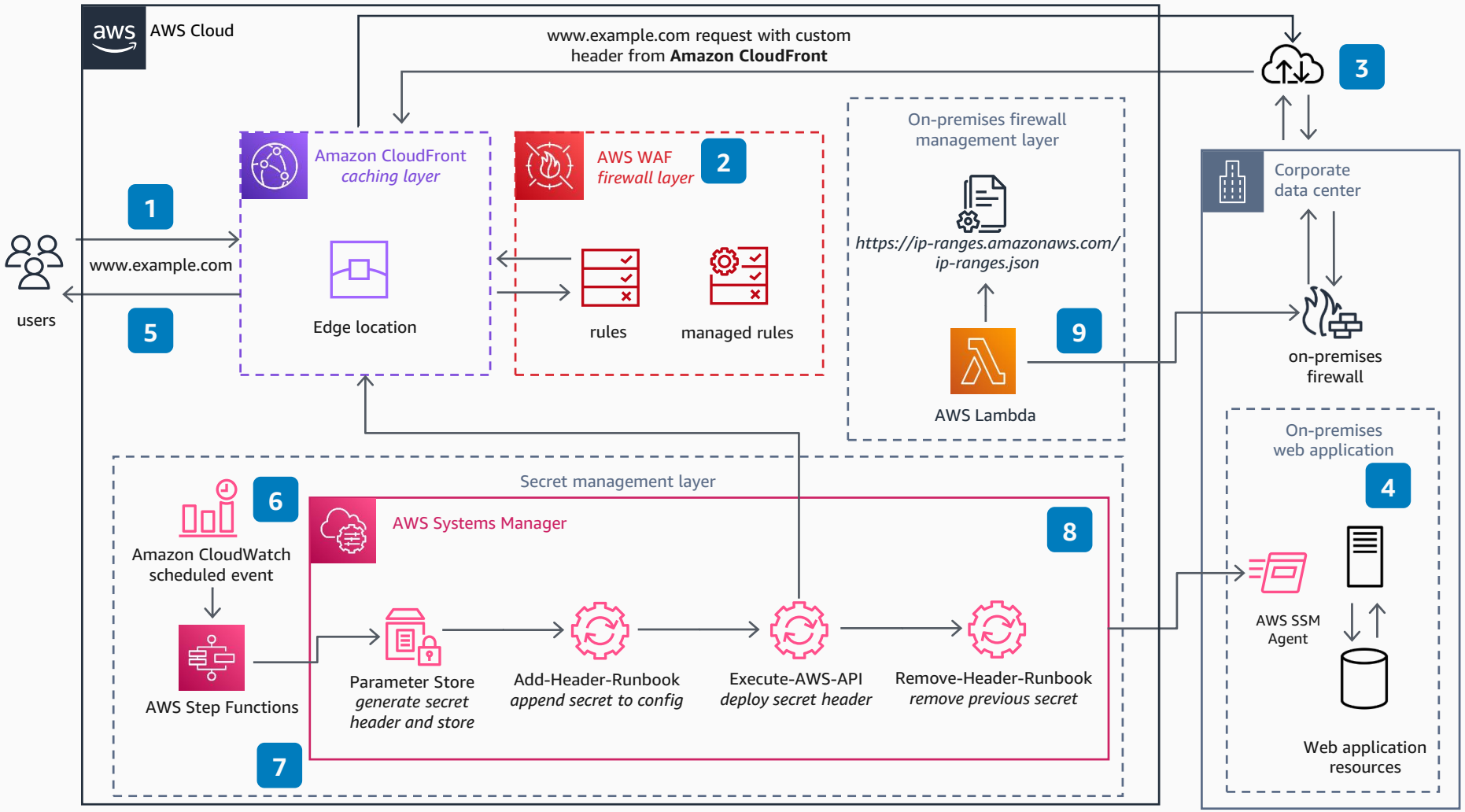
Securing Custom Origins with AWS WAF

This architecture shows how to protect any endpoint, including non-AWS endpoints, against common web vulnerabilities with AWS WAF by leveraging custom origins and custom secret headers in Amazon CloudFront.



1. Users make a request to the web application. Domain name system (DNS) records direct the user to the closest **Amazon CloudFront** edge location.
2. **AWS WAF** inspects the traffic by using both custom and managed rules for common web exploit attacks. Traffic is logged for future analysis and malicious traffic can be blocked.
3. **Amazon CloudFront** injects a secret custom header into the request and re-directs the request to the on-premises web application.
4. The web application is configured to drop or block any request that arrives without the secret custom header added by **Amazon CloudFront**. This ensures all traffic is inspected by **AWS WAF,** protecting the application from direct access.
5. Users receive the response to their request as normal from **Amazon CloudFront**. Data is then cached at the edge location for the next request.
6. The secret header rotation and deployment process is orchestrated by an **AWS Step Functions** workflow on a configurable schedule
7. The **AWS Step Functions** workflow generates a new secret for the **custom header** value and stores it in **AWS Parameter Store.**
8. The new header value is added to one or more web app servers via the **AWS Systems Manager Agent (AWS SSM Agent)** and **Automation Runbooks** using deployment strategies like rolling updates with error controls. When finalized, it deploys the new header to **Amazon CloudFront**. After waiting for propagation to all edge locations, the old secret is removed from the web apps
9. The on-premise firewall is updated to allow only the **AWS CloudFront** IP addresses to the web application as an additional protection layer to prevent direct access by users

Reviewed for technical accuracy July 26, 2022

AWS Reference Architecture