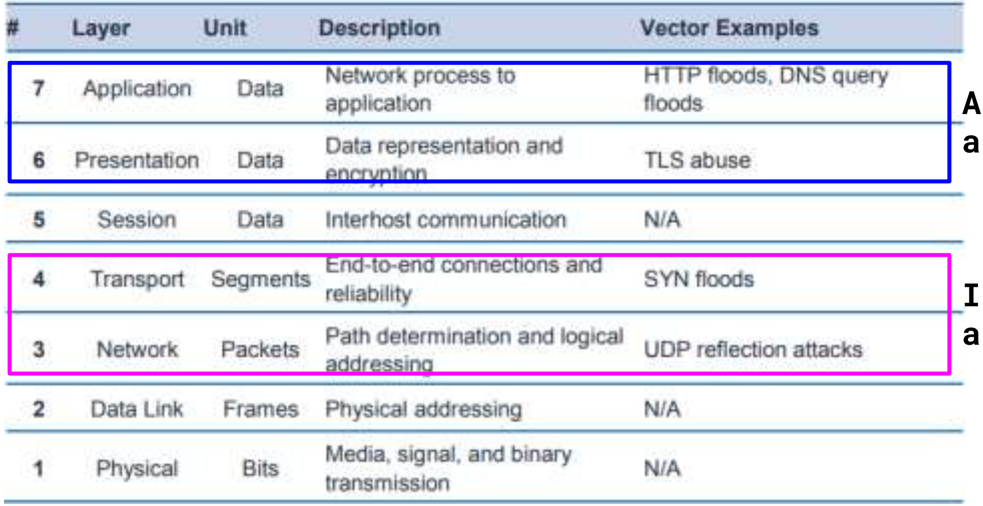
SAP-C01: Security, Identity & Compliance

IAM, WAF, Shield, KMS, CloudHSM

AWS Certified Solutions Architect– Professional (SAP-C01)
Study notes - Sep’2019

AWS Best Practices for DDoS Resiliency - Attacks

https://d1.awsstatic.com/whitepapers/Security/DDoS_White_Paper.pdf

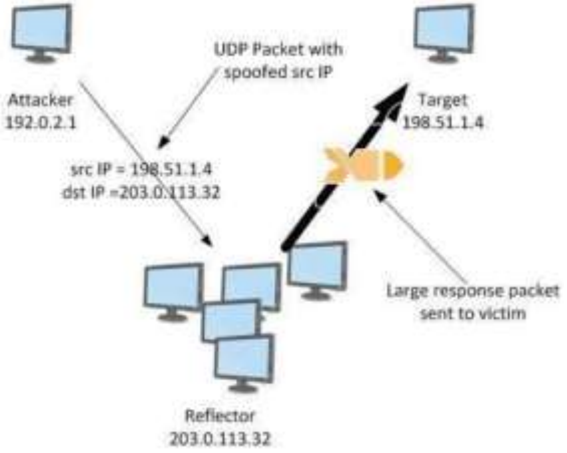


#	Layer	Unit	Description	Vector Examples
7	Application	Data	Network process to application	HTTP floods, DNS query floods
6	Presentation	Data	Data representation and encryption	TLS abuse
5	Session	Data	Interhost communication	N/A
4	Transport	Segments	End-to-end connections and reliability	SYN floods
3	Network	Packets	Path determination and logical addressing	UDP reflection attacks
2	Data Link	Frames	Physical addressing	N/A
1	Physical	Bits	Media, signal, and binary transmission	N/A

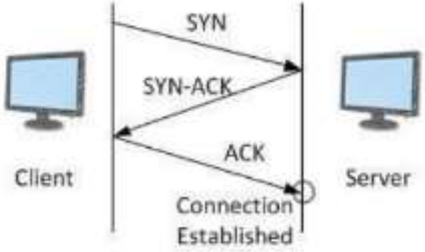
Application attacks

Infrastructure attacks

UDP Reflection attack
(Src IP spoofing)



SYN Floods
(No Final ACK)



HTTP floods	High volume seemingly valid requests
Cache busting	Variations in query string to circumvent CDN caching
DNS	Uses many well formed DNS queries to exhaust DN server
TLS Abuse	Send intelligible data, force re-negotiating the encryption method, Open many TLS sessions

AWS Best Practices for DDoS Resiliency - Mitigation

https://d1.awsstatic.com/whitepapers/Security/DDoS_White_Paper.pdf

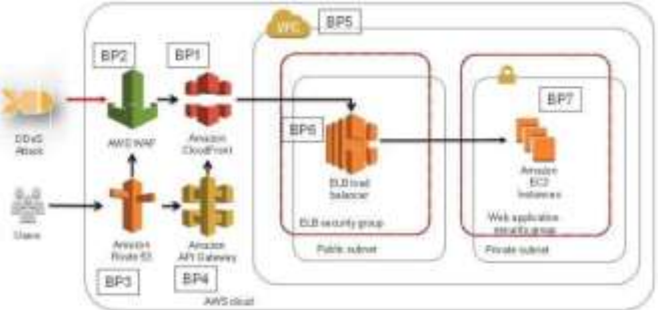
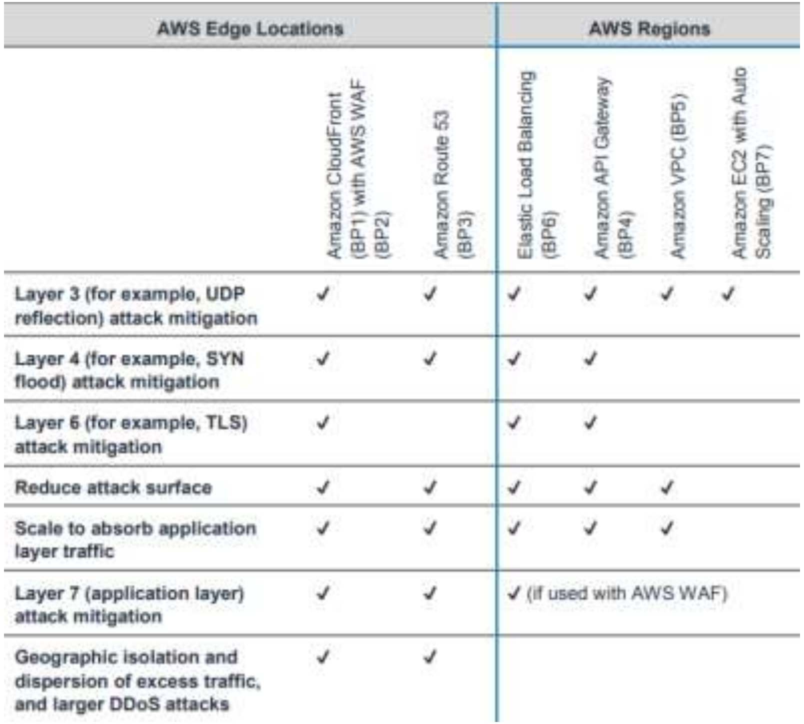


Figure 5: DDoS-resilient Reference Architecture

AWS Shield Standard → Available for all AWS services in all regions at no additional charge, ONLY PROTECTS AGAINST Layer 3 (Network) & 4 (Transport) attacks. DDoS attacks are detected by a system that automatically baselines traffic, identifies anomalies, and, as necessary, creates mitigations.

Use edge service like **CloudFront** & **Route53** for:

1. AWS Shield DDoS mitigation systems that are integrated with AWS edge services, reducing time-to-mitigate from minutes to sub-second.
2. Stateless SYN Flood mitigation techniques that proxy and verify incoming connections before passing them to the protected service.
3. Automatic traffic engineering systems that can disperse or isolate the impact of large volumetric DDoS attacks.
4. Application layer defense when combined with **AWS WAF** that does not require changing your current application architecture (for example, in an AWS Region or on-premises datacenter).



	AWS Edge Locations		AWS Regions			
	Amazon CloudFront (BP1) with AWS WAF (BP2)	Amazon Route 53 (BP3)	Elastic Load Balancing (BP6)	Amazon API Gateway (BP4)	Amazon VPC (BP5)	Amazon EC2 with 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	✓	✓	✓	✓	✓	
Layer 7 (application layer) attack mitigation	✓	✓	✓ (if used with AWS WAF)			
Geographic isolation and dispersion of excess traffic, and larger DDoS attacks	✓	✓				

AWS Best Practices for DDoS Resiliency

https://d1.awsstatic.com/whitepapers/Security/DDoS_White_Paper.pdf

AWS Shield Advanced →

1. This optional DDoS mitigation service helps you protect an application hosted on any AWS Region or hosted outside of AWS.
2. The service is available globally for Amazon CloudFront and Amazon Route 53.
3. It's also available in select AWS Regions for CLB,ALB, and EIPs. Using AWS Shield Advanced with EIPs allows you to protect NLBs or EC2 instances.

With AWS Shield Advanced, you get the following additional benefits:

- Access to the AWS DRT for assistance in mitigating DDoS attacks that impact application availability.
- DDoS attack visibility by using the AWS Management Console, API, and Amazon CloudWatch metrics and alarms.
- Access to the Global Threat Environment dashboard, which provides an overview of DDoS attacks observed and mitigated by AWS.
- Access to AWS WAF, at no additional cost, for the mitigation of application layer DDoS attacks (when used with CloudFront or ALB).
- Automatic baselining of web traffic attributes, when used with AWS WAF.
- Access to AWS Firewall Manager, at no additional cost, for automated policy enforcement. This service lets security administrators centrally control and manage AWS WAF rules.
- Sensitive detection thresholds which routes traffic into DDoS mitigation system earlier and can improve time-to-mitigate attacks against Amazon EC2 or NLB, when used with EIP.
- Cost protection that allows 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.

AWS Best Practices for DDoS Resiliency

https://d1.awsstatic.com/whitepapers/Security/DDoS_White_Paper.pdf

BP1	CloudFront	Use **AWS WAF with Web ACLs** to filter & block requests. Automatically **block the IP addresses** of bad actors **using IP match** conditions when requests matching a rule exceed a threshold that you define. Requests from offending client IP addresses will receive 403 Forbidden. Use **AWS Firewall manager** to manage WAF rules. Review your **web server logs** or use **AWS WAF's logging** and **Sampled Requests features.** **Cache** static content and dynamic content using variable TTL. It **accepts only well-formed requests** thus avoiding SYN floods and UDP reflection attacks. Use **OAI** to protect static content in S3. Use **Geo-restriction for whitelisting requests from specific countries. Use AWS DRT team** to enable rules for mitigation based on requests. Prevents **non-web traffic.** Can automatically **close connections from slow reading or slow writing** attackers.
BP2	CloudFront	
BP3	Route53	Uses techniques like **shuffle sharding** and **anycast striping**, that can help users access your application even if the DNS service is targeted by a DDoS attack.
BP4	API gateway	Recommendation is to **use Regional Api endpoint** (as against edge) enabled with **AWS WAF**. **Forward all headers** to API gateway to disable CF caching. Send the origin custom header "**x-api-key**" to protect APi gateway from direct access. Add **rate limits**.
BP5	Sgroups & NACLs	All internet traffic to a **security group** is implicitly denied unless you create an allow rule to permit the traffic. E.g use SG to allow VPC traffic only from Cfront. Also, use Edge-to-origin request headers such as "**X-Shared-Secret**" to help validate that requests made to your origin were sent from CloudFront. Use **NACLs** and specify both **allow and deny rules**. If your application is used only for TCP traffic, you can create a rule to deny all UDP traffic, or vice versa. If using **EIP**, register it as protected resource for Shield advanced.
BP6	Load balancing	For **web applications**, use **ALB** to route traffic based on its content and accept only well-formed web requests. This means that many common DDoS attacks, like SYN floods or UDP reflection attacks, will be blocked by ALB, protecting your application from the attack. For **TCP-based applications**, use **NLB** to route traffic to EC2 instances at ultra-low latency. When you create an NLB, a network interface is created for each AZ that you enable. You have the option to assign an EIP address per subnet enabled for the load balancer. One key consideration with NLB is that any traffic that reaches the load balancer on a valid listener will be routed to your EC2 instances, not absorbed.
BP7	EC2 size AWS regions	Resize compute capacity to scale up as reqd. Use 25 gigabit nw interfaces and Enhanced nw Use appropriate regions and AZs

AWS Best Practices for DDoS Resiliency - monitoring & Reacting

https://d1.awsstatic.com/whitepapers/Security/DDoS_White_Paper.pdf

1. 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. **Shield advanced metrics** - **DDoSDetected**, **DDoSAttackBitsPerSecond**, **DDoSAttackPacketsPerSecond**, or **DDoSAttackRequestsPerSecond**

2. 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 DRT. You can monitor these metrics by integrating CW with third-parties, such as Slack or PagerDuty.

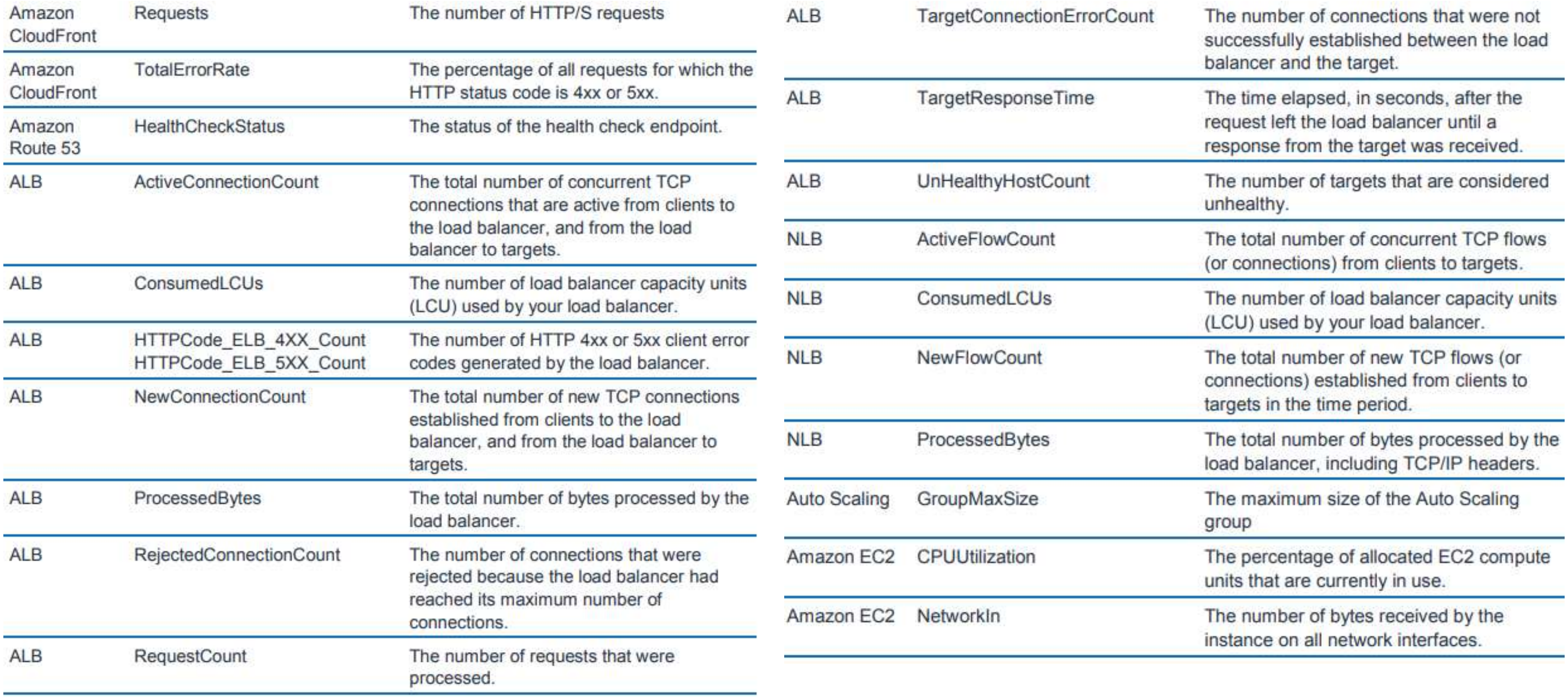
3. An application layer attack can elevate many CW metrics. If you're using **AWS WAF**, you can use CW to monitor and alarm on increases in requests that you've set in WAF to be allowed **WAF metrics** (**AllowedRequests**), counted (**CountedRequests**), or blocked (**BlockedRequests**). This allows you to receive a notification if the level of traffic exceeds what your application can handle.

4. 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 summary and details about attacks that have been detected. In addition, the Global Threat Environment Dashboard provides summary information about all DDoS attacks that have been detected by AWS.

5. Another tool that can help you gain visibility into traffic that is targeting your application is **VPC Flow Logs**. Each flow log record includes the following: source and destination IP addresses, source and destination ports, protocol, and the number of packets and bytes transferred during the capture window. You can use this information to help identify anomalies in network traffic and to identify a specific attack vector

AWS Best Practices for DDoS Resiliency - Metrics

https://d1.awsstatic.com/whitepapers/Security/DDoS_White_Paper.pdf



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.
ALB	ActiveConnectionCount	The total number of concurrent TCP connections that are active from clients to the load balancer, and from the load balancer to targets.
ALB	ConsumedLCUs	The number of load balancer capacity units (LCU) used by your load balancer.
ALB	HTTPCode_ELB_4XX_Count HTTPCode_ELB_5XX_Count	The number of HTTP 4xx or 5xx client error codes generated by the load balancer.
ALB	NewConnectionCount	The total number of new TCP connections established from clients to the load balancer, and from the load balancer to targets.
ALB	ProcessedBytes	The total number of bytes processed by the load balancer.
ALB	RejectedConnectionCount	The number of connections that were rejected because the load balancer had reached its maximum number of connections.
ALB	RequestCount	The number of requests that were processed.
ALB	TargetConnectionErrorCount	The number of connections that were not successfully established between the load balancer and the target.
ALB	TargetResponseTime	The time elapsed, in seconds, after the request left the load balancer until a response from the target was received.
ALB	UnHealthyHostCount	The number of targets that are considered unhealthy.
NLB	ActiveFlowCount	The total number of concurrent TCP flows (or connections) from clients to targets.
NLB	ConsumedLCUs	The number of load balancer capacity units (LCU) used by your load balancer.
NLB	NewFlowCount	The total number of new TCP flows (or connections) established from clients to targets in the time period.
NLB	ProcessedBytes	The total number of bytes processed by the load balancer, 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.

VPC Flow logs

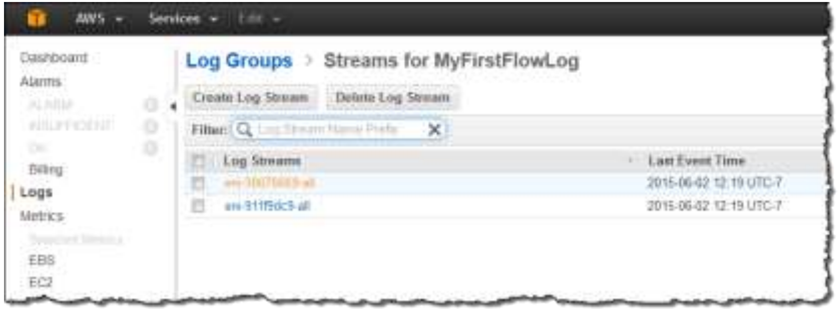
https://aws.amazon.com/blogs/aws/vpc-flow-logs-log-and-view-network-traffic-flows/

Once enabled for a particular VPC, VPC subnet, or ENI, relevant network traffic will be logged to CloudWatch Logs. You can create alarms that will fire if certain types of traffic are detected; you can also create metrics to help you to identify trends and patterns.

The information captured includes information about allowed and denied traffic (based on security group and network ACL rules). Each flow log record includes the following: **source and destination IP addresses**, **source** and **destination ports**, **IANA protocol**, and the **number of packets and bytes transferred** during the capture window and an **action (ACCEPT or REJECT)**.

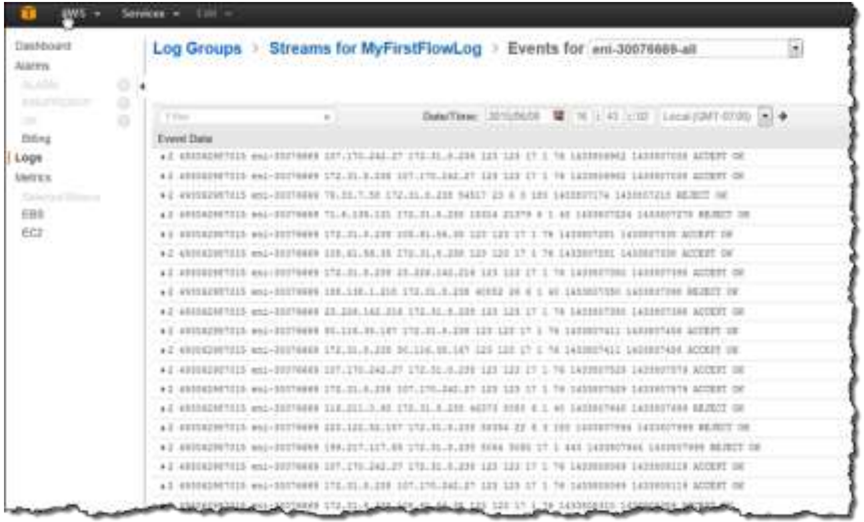
Each CW log group will contain a separate stream for each ENI. Each stream, in turn, contains a series of flow log records. The Flow Logs will not include any of the following traffic:

1. Traffic to Amazon DNS servers, including queries for private hosted zones.
2. Windows license activation traffic for licenses provided by Amazon.
3. Requests for instance metadata.
4. DHCP requests or responses.



Log Groups › Streams for MyFirstFlowLog

Log Streams	Last Event Time
eni-30076669-all	2015-06-02 12:19 UTC-7
eni-311f5dc3-all	2015-06-02 12:19 UTC-7



Log Groups › Streams for MyFirstFlowLog › Events for eni-30076669-all

```
Event Data
2 [illegible] eni-30076669 107.170.242.27 172.31.8.238 123 123 17 1 76 [illegible] [illegible] ACCEPT OK
2 [illegible] eni-30076669 172.31.8.238 107.170.242.27 123 123 17 1 76 [illegible] [illegible] ACCEPT OK
2 [illegible] eni-30076669 78.31.7.58 172.31.8.238 54517 23 6 3 180 [illegible] [illegible] REJECT OK
2 [illegible] eni-30076669 71.6.135.131 172.31.8.238 10034 21379 6 1 40 [illegible] [illegible] REJECT OK
2 [illegible] eni-30076669 172.31.8.238 108.61.56.35 123 123 17 1 76 [illegible] [illegible] ACCEPT OK
2 [illegible] eni-30076669 108.61.56.35 172.31.8.238 123 123 17 1 76 [illegible] [illegible] ACCEPT OK
2 [illegible] eni-30076669 172.31.8.238 23.228.142.218 123 123 17 1 76 [illegible] [illegible] ACCEPT OK
2 [illegible] eni-30076669 188.138.1.210 172.31.8.238 40552 20 6 1 40 [illegible] [illegible] REJECT OK
2 [illegible] eni-30076669 23.228.142.218 172.31.8.238 123 123 17 1 76 [illegible] [illegible] ACCEPT OK
2 [illegible] eni-30076669 96.116.95.187 172.31.8.238 123 123 17 1 76 [illegible] [illegible] ACCEPT OK
2 [illegible] eni-30076669 172.31.8.238 96.116.95.187 123 123 17 1 76 [illegible] [illegible] ACCEPT OK
2 [illegible] eni-30076669 107.170.242.27 172.31.8.238 123 123 17 1 76 [illegible] [illegible] ACCEPT OK
2 [illegible] eni-30076669 172.31.8.238 107.170.242.27 123 123 17 1 76 [illegible] [illegible] ACCEPT OK
2 [illegible] eni-30076669 118.211.3.80 172.31.8.238 46073 3389 6 1 40 [illegible] [illegible] REJECT OK
2 [illegible] eni-30076669 222.122.52.157 172.31.8.238 58594 22 6 3 180 [illegible] [illegible] REJECT OK
2 [illegible] eni-30076669 199.217.117.85 172.31.8.238 5666 5060 17 1 443 [illegible] [illegible] REJECT OK
2 [illegible] eni-30076669 107.170.242.27 172.31.8.238 123 123 17 1 76 [illegible] [illegible] ACCEPT OK
2 [illegible] eni-30076669 172.31.8.238 107.170.242.27 123 123 17 1 76 [illegible] [illegible] ACCEPT OK
```

AWS Inspector

https://docs.aws.amazon.com/inspector/latest/userguide/inspector_introduction.html

Amazon Inspector tests the network accessibility of your Amazon EC2 instances and the security state of your applications that run on those instances. Amazon Inspector assesses applications for exposure, vulnerabilities, and deviations from best practices. After performing an assessment, Amazon Inspector produces a detailed list of security findings that is organized by level of severity.

With Amazon Inspector, you can automate security vulnerability assessments throughout your development and deployment pipelines or for static production systems. This allows you to make security testing a regular part of development and IT operations.

Amazon Inspector also offers predefined software called an *agent* that you can optionally install in the operating system of the EC2 instances that you want to assess. The agent monitors the behavior of the EC2 instances, including network, file system, and process activity. It also collects a wide set of behavior and configuration data (telemetry).

Features of Amazon Inspector

- **Configuration scanning and activity monitoring engine** – Amazon Inspector provides an agent that analyzes system and resource configuration. It also monitors activity to determine what an assessment target looks like, how it behaves, and its dependent components. The combination of this telemetry provides a complete picture of the target and its potential security or compliance issues.

- **Built-in content library** – Amazon Inspector includes a built-in library of rules and reports. These include checks against best practices, common compliance standards, and vulnerabilities. The checks include detailed recommended steps for resolving potential security issues.

- **Automation through an API** – Amazon Inspector can be fully automated through an API. This allows you to incorporate security testing into the development and design process, including selecting, executing, and reporting the results of those tests.

Trusted Advisor gives counsel about your AWS Account in the regions of:

Cost Optimization, Fault Tolerance, Performance, Service Limits, Security

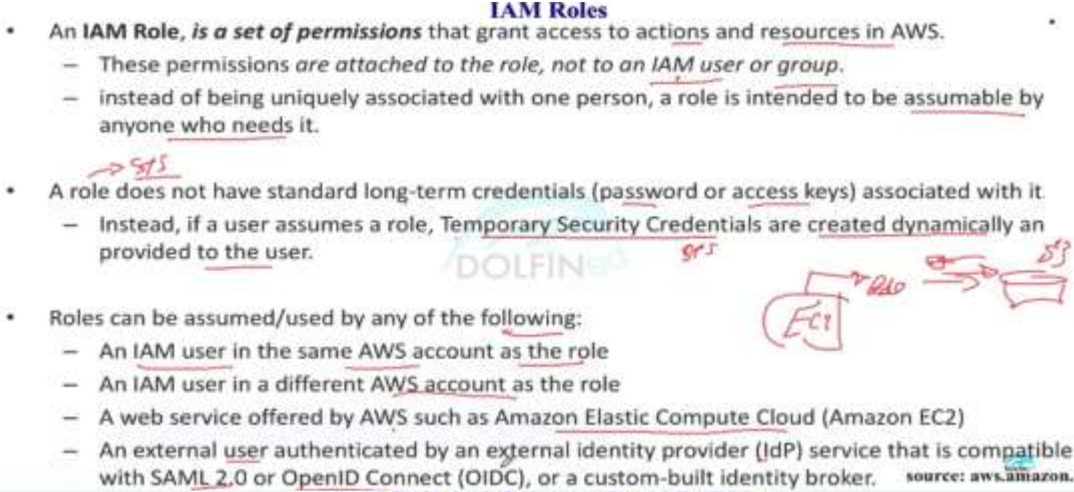
Trusted Advisor applies to the AWS account and AWS administrations, help customers follow general AWS best practices.

Amazon Inspector applies to the content of various EC2 occurrences. It is mostly about what happens on an instance: does the software conform to various best practises, patches installed etc.

Aws config rules are about resources. You can’t check what’s going on inside an instance but you could check that it’s running approved ami, has all of its volumes encrypted etc. There’s checks you can setup against lots of different types of resources.

IAM

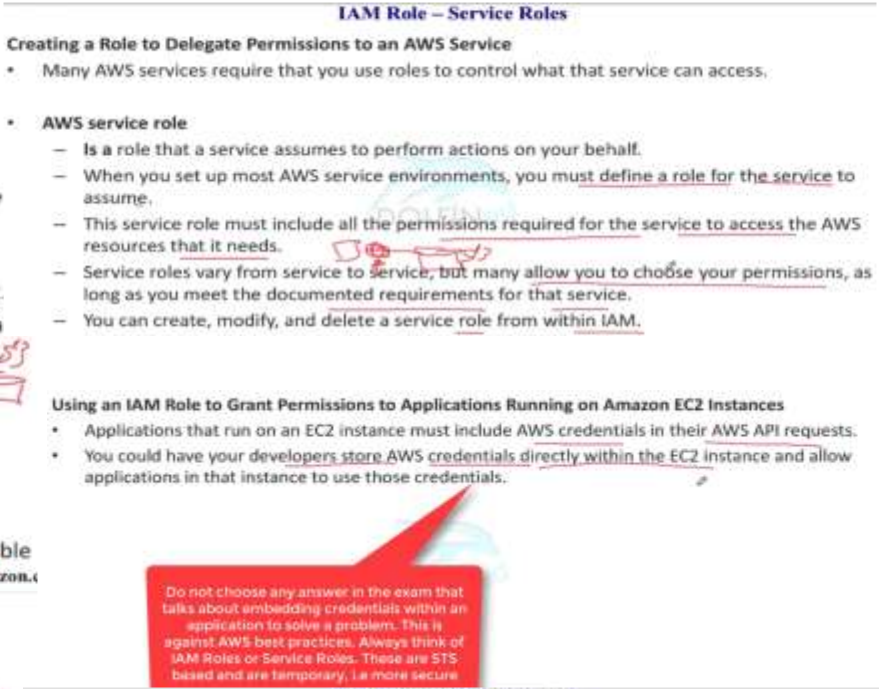
IAM Roles



IAM Roles

- An **IAM Role**, ***is a set of permissions*** that grant access to actions and resources in AWS.
  - These permissions *are attached to the role, not to an IAM user or group*.
  - instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it.
- A role does not have standard long-term credentials (password or access keys) associated with it
  - Instead, if a user assumes a role, Temporary Security Credentials are created dynamically an provided to the user.
- Roles can be assumed/used by any of the following:
  - An IAM user in the same AWS account as the role
  - An IAM user in a different AWS account as the role
  - A web service offered by AWS such as Amazon Elastic Compute Cloud (Amazon EC2)
  - An external user authenticated by an external identity provider (IdP) service that is compatible with SAML 2.0 or OpenID Connect (OIDC), or a custom-built identity broker.

source: aws.amazon.c



IAM Role – Service Roles

Creating a Role to Delegate Permissions to an AWS Service

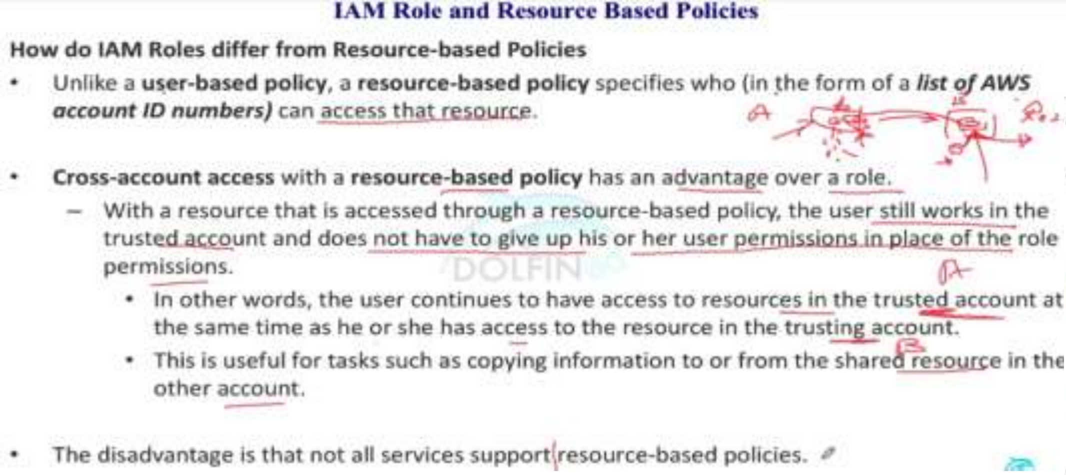
- Many AWS services require that you use roles to control what that service can access.

- **AWS service role**
  - **Is a** role that a service assumes to perform actions on your behalf.
  - When you set up most AWS service environments, you must define a role for the service to assume.
  - This service role must include all the permissions required for the service to access the AWS resources that it needs.
  - Service roles vary from service to service, but many allow you to choose your permissions, as long as you meet the documented requirements for that service.
  - You can create, modify, and delete a service role from within IAM.

Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances

- Applications that run on an EC2 instance must include AWS credentials in their AWS API requests.
- You could have your developers store AWS credentials directly within the EC2 instance and allow applications in that instance to use those credentials.

Do not choose any answer in the exam that talks about embedding credentials within an application to solve a problem. This is against AWS best practices. Always think of IAM Roles or Service Roles. These are STS based and are temporary, i.e more secure



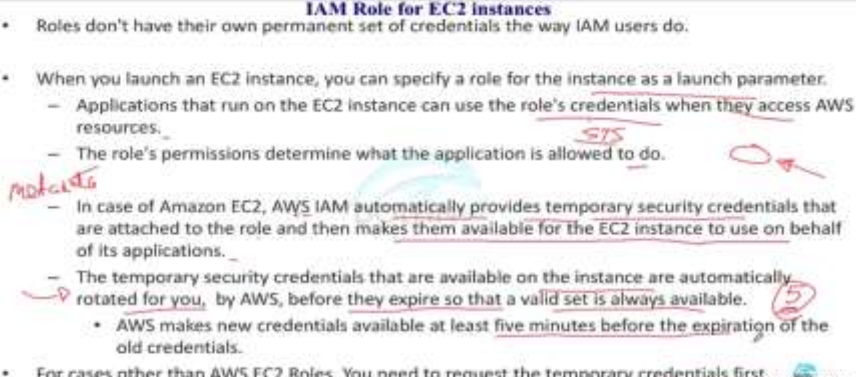
IAM Role and Resource Based Policies

How do IAM Roles differ from Resource-based Policies

- Unlike a **user-based policy**, a **resource-based policy** specifies who (in the form of a ***list of AWS account ID numbers)*** can access that resource.

- **Cross-account access** with a **resource-based policy** has an advantage over a role.
  - With a resource that is accessed through a resource-based policy, the user still works in the trusted account and does not have to give up his or her user permissions in place of the role permissions.
    - In other words, the user continues to have access to resources in the trusted account at the same time as he or she has access to the resource in the trusting account.
    - This is useful for tasks such as copying information to or from the shared resource in the other account.

- The disadvantage is that not all services support resource-based policies.

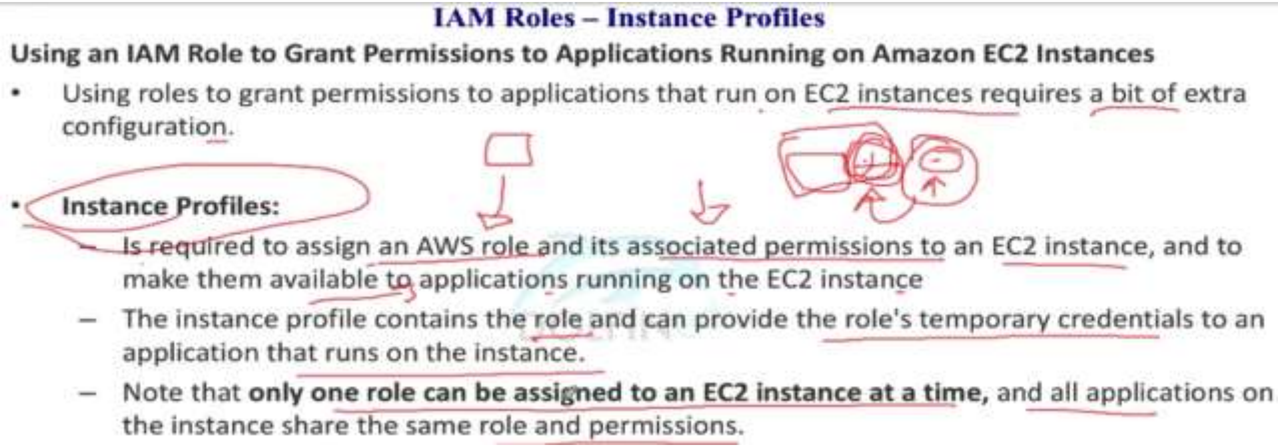


IAM Role for EC2 instances

- Roles don't have their own permanent set of credentials the way IAM users do.

- When you launch an EC2 instance, you can specify a role for the instance as a launch parameter.
  - Applications that run on the EC2 instance can use the role's credentials when they access AWS resources.
  - The role's permissions determine what the application is allowed to do.
  - In case of Amazon EC2, AWS IAM automatically provides temporary security credentials that are attached to the role and then makes them available for the EC2 instance to use on behalf of its applications.
  - The temporary security credentials that are available on the instance are automatically rotated for you, by AWS, before they expire so that a valid set is always available.
    - AWS makes new credentials available at least five minutes before the expiration of the old credentials.
- For cases other than AWS EC2 Roles, You need to request the temporary credentials first

IAM - EC2 Instance Profiles



IAM Roles – Instance Profiles

Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances

- Using roles to grant permissions to applications that run on EC2 instances requires a bit of extra configuration.

- **Instance Profiles:**
  - Is required to assign an AWS role and its associated permissions to an EC2 instance, and to make them available to applications running on the EC2 instance
  - The instance profile contains the role and can provide the role's temporary credentials to an application that runs on the instance.
  - Note that **only one role can be assigned to an EC2 instance at a time,** and all applications on the instance share the same role and permissions.

Roles are designed to be “assumed” by other principals which do define “who am I?”, such as users, Amazon services, and EC2 instances. An instance profile, on the other hand, defines “who am I?” Just like an IAM user represents a person, an instance profile represents EC2 instances. The only permissions an EC2 instance profile has is the power to assume a role.

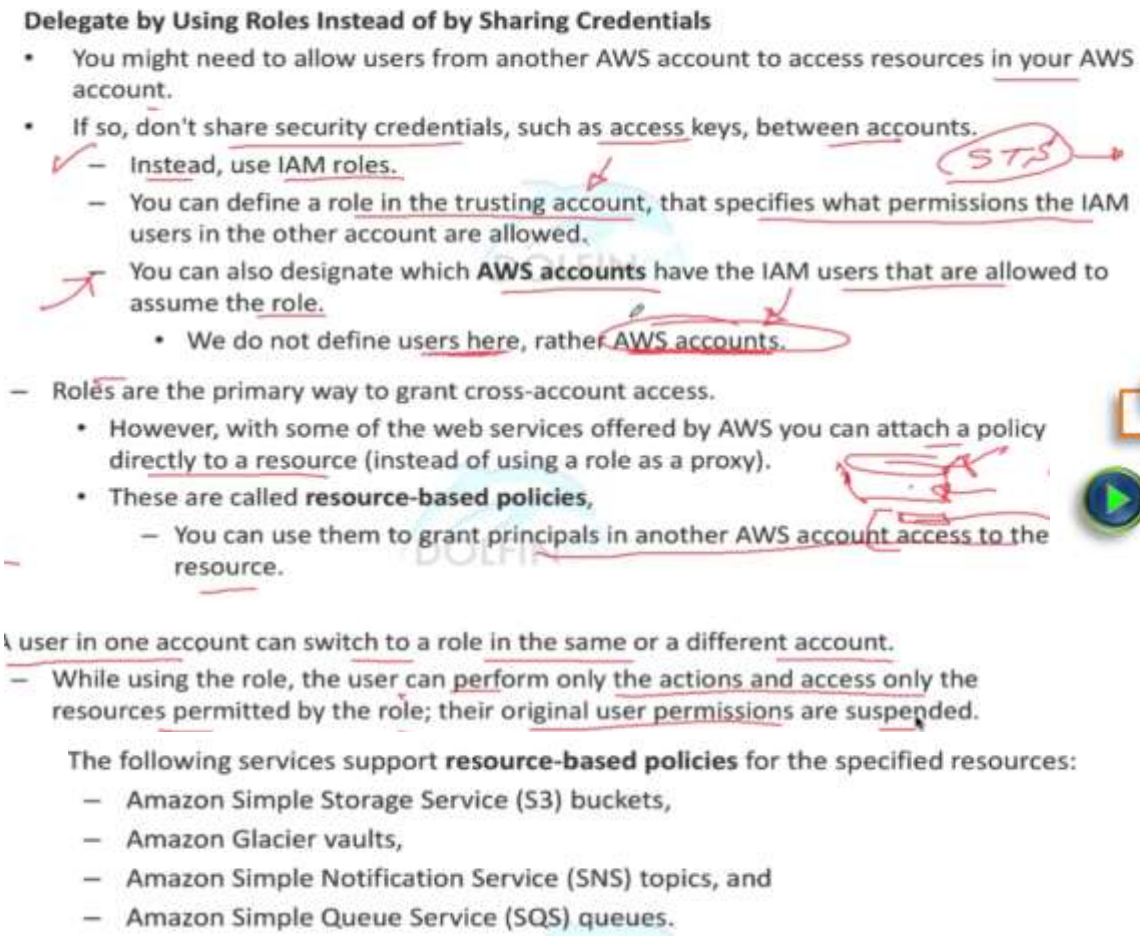
So the EC2 instance runs under the EC2 instance profile, defining “who” the instance is. It then “assumes” the IAM role, which ultimately gives it any real power.

When you create an IAM Role for EC2 using the AWS Console, it creates both an EC2 instance profile as well as an IAM role. However, if you are using the AWS CLI, SDKs, or CloudFormation, you will need to explicitly define both:

- An IAM role with policies and permissions, and
- An EC2 instance profile specifying which roles it can assume

E.g for EBeanStalk, the service role is assumed by EBS to use other AWS services on your behalf. The instance profile is applied to the instances in your environment and allows them to retrieve application versions from Amazon Simple Storage Service (Amazon S3), upload logs to Amazon S3, and perform other tasks that vary depending on the environment type and platform.

IAM - Cross Account Access



Delegate by Using Roles Instead of by Sharing Credentials

- You might need to allow users from another AWS account to access resources in your AWS account.
- If so, don't share security credentials, such as access keys, between accounts. STS
  - Instead, use IAM roles.
  - You can define a role in the trusting account, that specifies what permissions the IAM users in the other account are allowed.
  - You can also designate which **AWS accounts** have the IAM users that are allowed to assume the role.
    - We do not define users here, rather AWS accounts.

- Roles are the primary way to grant cross-account access.
  - However, with some of the web services offered by AWS you can attach a policy directly to a resource (instead of using a role as a proxy).
  - These are called **resource-based policies**,
    - You can use them to grant principals in another AWS account access to the resource.

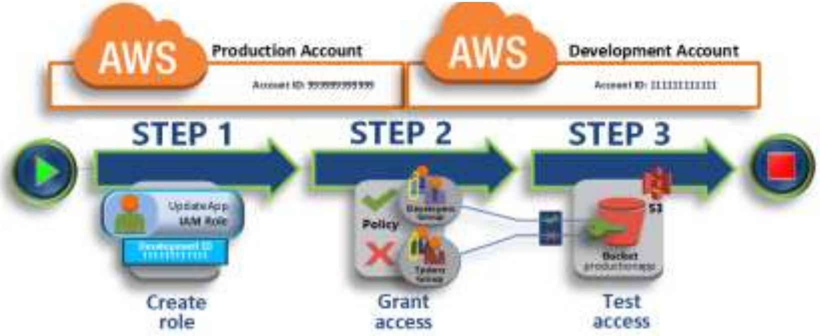
A user in one account can switch to a role in the same or a different account.

- While using the role, the user can perform only the actions and access only the resources permitted by the role; their original user permissions are suspended.

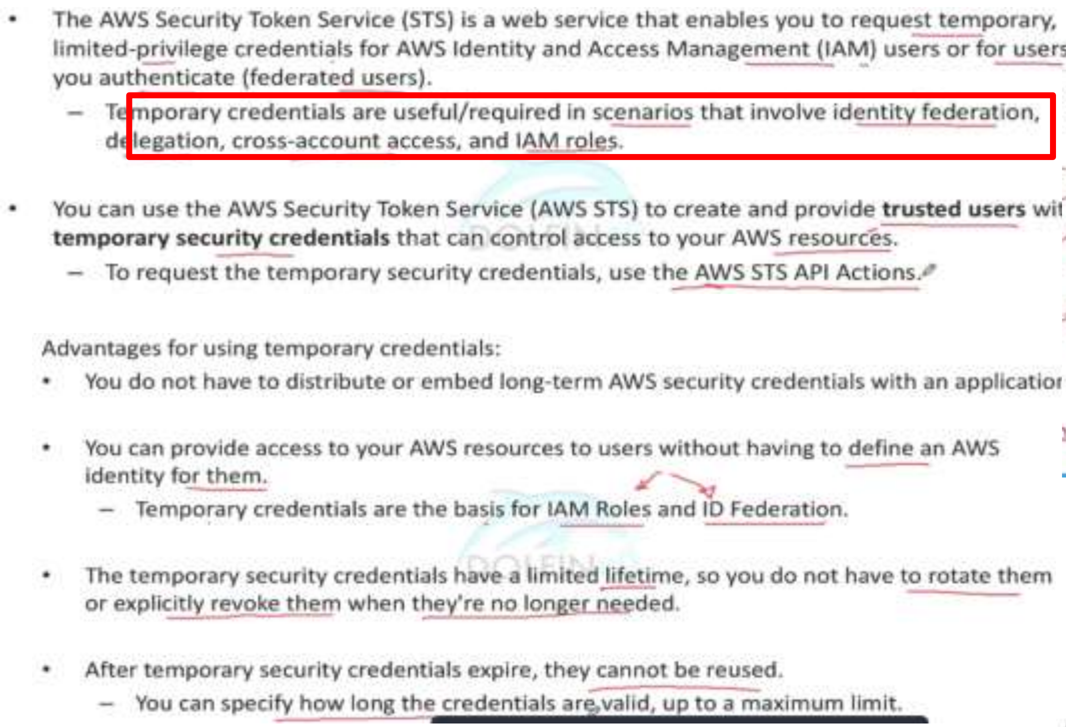
The following services support **resource-based policies** for the specified resources:

- Amazon Simple Storage Service (S3) buckets,
- Amazon Glacier vaults,
- Amazon Simple Notification Service (SNS) topics, and
- Amazon Simple Queue Service (SQS) queues.

Use a role to delegate access to resources that are in different AWS accounts



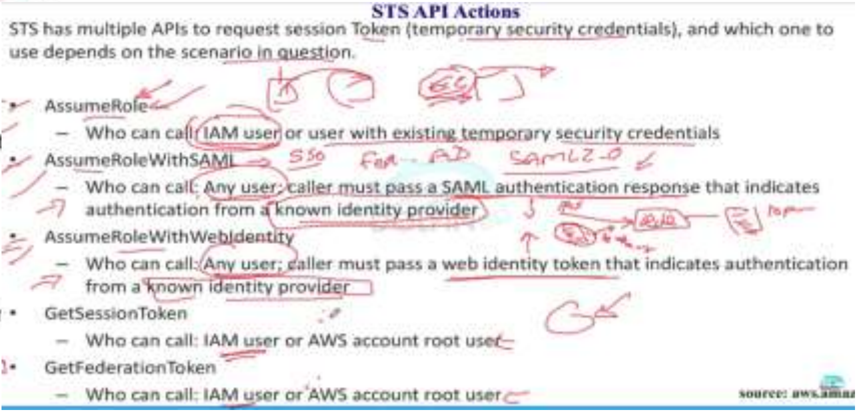
IAM - STS



- The AWS Security Token Service (STS) is a web service that enables you to request temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) users or for users that you authenticate (federated users).
  - Temporary credentials are useful/required in scenarios that involve identity federation, delegation, cross-account access, and IAM roles.
- You can use the AWS Security Token Service (AWS STS) to create and provide **trusted users wit temporary security credentials** that can control access to your AWS resources.
  - To request the temporary security credentials, use the AWS STS API Actions.

Advantages for using temporary credentials:

- You do not have to distribute or embed long-term AWS security credentials with an applicatior
- You can provide access to your AWS resources to users without having to define an AWS identity for them.
  - Temporary credentials are the basis for IAM Roles and ID Federation.
- The temporary security credentials have a limited lifetime, so you do not have to rotate them or explicitly revoke them when they're no longer needed.
- After temporary security credentials expire, they cannot be reused.
  - You can specify how long the credentials are valid, up to a maximum limit.

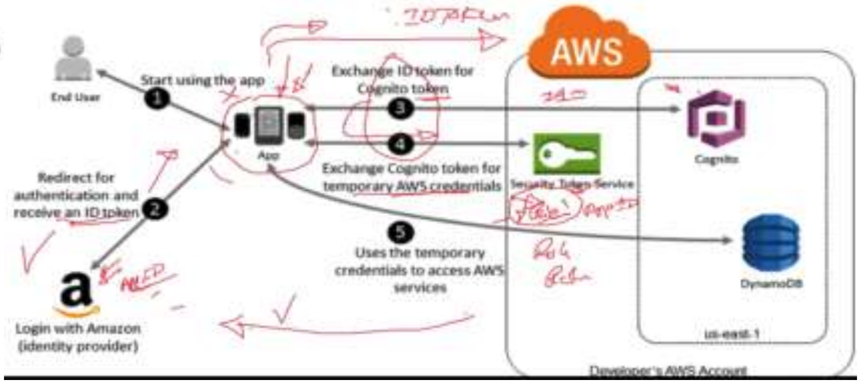


STS API Actions

STS has multiple APIs to request session Token (temporary security credentials), and which one to use depends on the scenario in question.

- AssumeRole
  - Who can call: IAM user or user with existing temporary security credentials
- AssumeRoleWithSAML → SSO Fed - AD SAML2.0
  - Who can call: Any user; caller must pass a SAML authentication response that indicates authentication from a known identity provider
- AssumeRoleWithWebIdentity
  - Who can call: Any user; caller must pass a web identity token that indicates authentication from a known identity provider
- GetSessionToken
  - Who can call: IAM user or AWS account root user
- GetFederationToken
  - Who can call: IAM user or AWS account root user

source: aws.amaz


AWS
End User
1
Start using the app
App
2
Redirect for authentication and receive an ID token
Login with Amazon (identity provider)
3
Exchange ID token for Cognito token
4
Exchange Cognito token for temporary AWS credentials
5
Uses the temporary credentials to access AWS services
Cognito
Security Token Service
DynamoDB
us-east-1
Developer's AWS Account

IAM - SAML Federation used for enterprise/corporate integrations

- **Security Assertion Markup Language 2.0 (SAML 2.0)** is a version of the SAML open standard for exchanging authentication and authorization data between security domains.
  - SAML 2.0 is an XML –based Protocol that used Security Tokens containing Assertions to pass information about a principal (usually an end user) between a SAML authority, named an Identity Provider, and a SAML consumer, named a Service Provider.
  - SAML 2.0 enables web-based, cross-domain Single Sign-On (SSO), which helps reduce the administrative overhead of distributing multiple authentication tokens to the user.
- Using SAML, AWS enables federated single sign-on (SSO), which lets users sign into the AWS Management Console or make programmatic calls to AWS APIs by using assertions from a SAML-compliant identity provider (IdP) like ADFS.
- Enterprise Federation to AWS is possible using Windows Active Directory (AD), Active Directory Federation Services (ADFS) 2.0, and Security Assertion Markup Language 2.0.
  - Windows server includes ADFS, hence, it makes sense to use ADFS as your IdP.

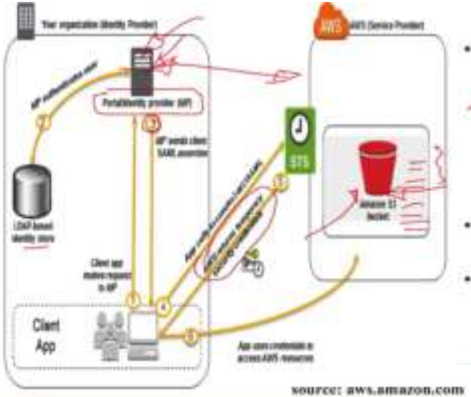
source: aws.amaz

Identity Federation with STS can be done through different APIs, the one used with SAML is

- **AssumeRoleWithSAML—**
  - Federation Through an Enterprise Identity Provider Compatible with SAML 2.0 Security Assertion Markup Language 2.0 (SAML)
- You can use single sign-on (SSO) to sign in to all of your SAML-enabled applications by using a single set of credentials.
- By enabling SAML authentication, you also can manage access to your applications centrally.
- SAML-enabled applications delegate authentication requests to your corporate directory. When users are removed from your directory, they are no longer able to sign in.

Accessing an AWS Service from On-prem

1. A user in your organization uses a client app to request authentication from your organization's IdP. This is done through the IdP's sign-in page
2. The IdP authenticates the user against your organization's identity store.
3. The IdP constructs a SAML assertion with information (authentication response) about the user and sends the assertion to the client app.
4. The client app calls the AWS STS AssumeRoleWithSAML API, passing the ARN of the SAML provider, the ARN of the role to assume, and the SAML assertion from IdP.
5. The API response to the client app includes temporary security credentials.
6. The client app uses the temporary security credentials to call Amazon S3 APIs.

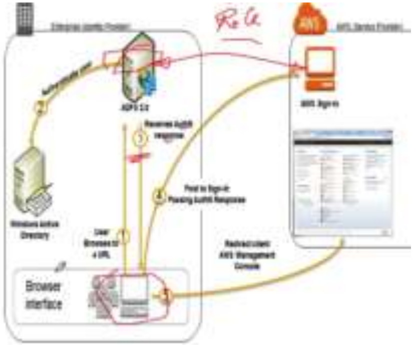


Your organization (Identity Provider)
AWS
AWS (Service Provider)
Portal/Identity provider (IdP)
IdP authenticates user
IdP sends client SAML assertion
LDAP-based identity store
Client app makes request to IdP
Client App
STS
Amazon S3 bucket
App uses credentials to access AWS resources
1
2
3
4
5

source: aws.amazon.com

Accessing AWS Console from On-prem (SSO)

- The user's browser receives a SAML assertion in the form of an authentication response from ADFS.
- The browser posts the SAML assertion to the AWS sign-in endpoint for SAML (https://signin.aws.amazon.com/saml). Behind the scenes, sign-in uses the AssumeRoleWithSAML API to request temporary security credentials and then constructs a sign-in URL for the AWS Management Console.
- The browser receives the AWS sign-in URL and is redirected to the console.
- From the user's perspective, the process happens transparently. The user starts at an internal web site and ends up at the AWS Management Console, without IAM sing on or IAM credentials.



source: aws.ama

IAM - Scenarios

Person identity → Use IAM user credentials (Access key & Access secret)

Non-person (role) identities:

- Mobile Apps → Web Identity Federation (AssumeRoleWithWebIdentity)
- Corporate Sign in → SAML 2.0 Federated Access (AssumeRoleWithSAML)
- Corporate Sign in without SAML → AssumeRole or GetFederationToken
- EC2 accessing AWS services → AssumeRole (Instance Profile)
- Need to send MFA code → GetSessionToken

Providing Access to Externally Authenticated Users (Identity Federation)

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_common-scenarios_federated-users.html

Your users might already have identities outside of AWS, such as in your corporate directory. If those users need to work with AWS resources (or work with applications that access those resources), then those users also need AWS security credentials. You can use an IAM role to specify permissions for users whose identity is federated from your organization or a third-party identity provider (IdP).

1. **Federating Users of a Mobile or Web-based App with Amazon Cognito**

If you create a mobile or web-based app that accesses AWS resources, the app needs security credentials in order to make programmatic requests to AWS. For most mobile application scenarios, we recommend that you use Amazon Cognito. Amazon Cognito also provides API operations for synchronizing user data so that it is preserved as users move between devices.

1. **Federating Users with Public Identity Service Providers or OpenID Connect**

Whenever possible, use Cognito for mobile and web-based application scenarios. Cognito does most of the behind-the-scenes work with public identity provider services for you. It works with the same third-party services and also supports anonymous sign-ins. However, for more advanced scenarios, you can work directly with a third-party service like Login with Amazon, Facebook, Google, or any IdP that is compatible with OpenID Connect (OIDC).

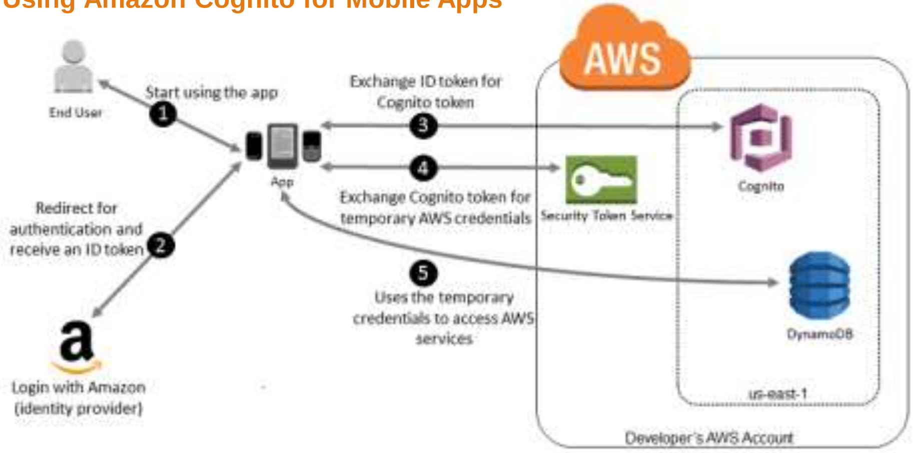
1. **Federating users with SAML 2.0**

If your organization already uses an identity provider software package that supports SAML 2.0 , you can create trust between your organization as an IdP and AWS as the service provider. You can then use SAML to provide your users with federated SSO to the AWS Mgmt Console or federated access to call AWS API operations.

1. **Federating users by creating a custom identity broker application**

If your identity store is not compatible with SAML 2.0, then you can build a custom identity broker application to perform a similar function. The broker application authenticates users, requests temporary credentials for users from AWS, and then provides them to the user to access AWS resources.

Using Amazon Cognito for Mobile Apps


End User
1
Start using the app
App
2
Redirect for authentication and receive an ID token
Login with Amazon (identity provider)
3
Exchange ID token for Cognito token
4
Exchange Cognito token for temporary AWS credentials
5
Uses the temporary credentials to access AWS services
AWS
Cognito
Security Token Service
DynamoDB
us-east-1
Developer's AWS Account

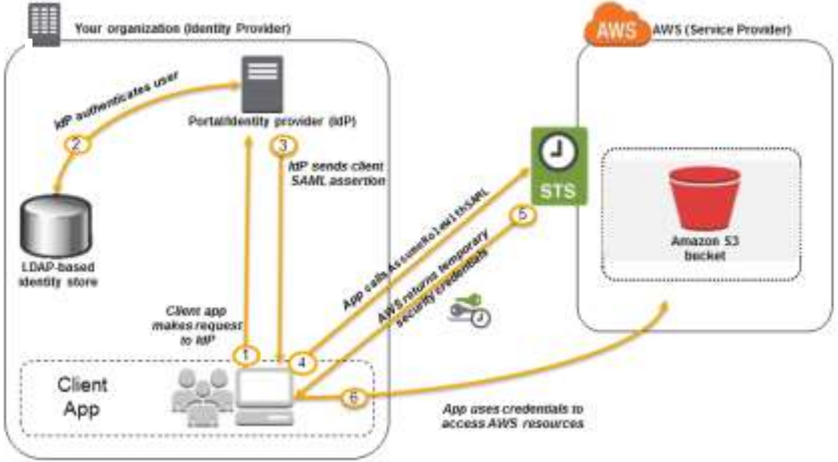
Identifying Users with Web Identity Federation

```
    "Effect": "Allow",
    "Action": ["s3:ListBucket"],
    "Resource": ["arn:aws:s3:::myBucket"],
    "Condition": {"StringLike": {"s3:prefix":
["Amazon/mynumbersgame/${www.amazon.com:user_id}/*"]}}
```

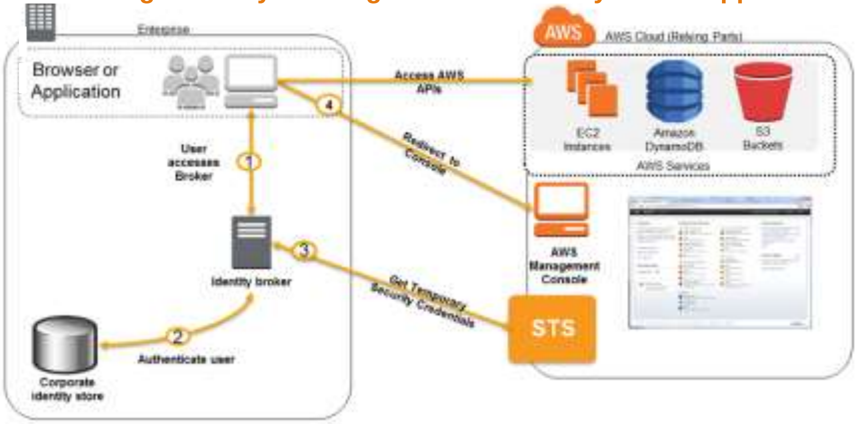
Uniquely Identifying Users in SAML-Based Federation

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:PutObject",
      "s3:DeleteObject"
    ],
    "Resource": [
   "arn:aws:s3:::exampleorgBucket/backup/${saml:namequalifier}/${saml:sub}",
"arn:aws:s3:::exampleorgBucket/backup/${saml:namequalifier}/${saml:sub}/*"
```

Using SAML-Based Federation for API Access to AWS


Your organization (Identity Provider)
AWS
AWS (Service Provider)
Portal/identity provider (IdP)
2
IdP authenticates user
LDAP-based identity store
3
IdP sends client SAML assertion
1
Client app makes request to IdP
Client App
4
App calls AssumeRoleWithSAML
5
AWS returns temporary security credentials
STS
Amazon S3 bucket
6
App uses credentials to access AWS resources

Federating users by creating a custom identity broker application



Requesting Temporary Security Credentials

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_request.html

AssumeRole—Cross-Account Delegation and Federation Through a Custom Identity Broker

The AssumeRole API operation is useful for allowing existing IAM users to access AWS resources that they don't already have access to, such as resources in another AWS account. It is also useful as a means to temporarily gain privileged access—for example, to provide multi-factor authentication (MFA). You must call this API using existing IAM user credentials.

AssumeRoleWithWebIdentity—Federation Through a Web-Based Identity Provider

The AssumeRoleWithWebIdentity API operation returns a set of temporary security credentials for federated users who are authenticated through a public identity provider. E.g. Amazon, Facebook, Google, or any OpenID Connect (OIDC)-compatible identity provider. This operation is useful for creating mobile applications or client-based web applications that require access to AWS. Using this operation means that your users do not need their own AWS or IAM identities.

AssumeRoleWithSAML—Federation Through an Enterprise Identity Provider Compatible with SAML 2.0

The AssumeRoleWithSAML API operation returns a set of temporary security credentials for federated users who are authenticated by your organization's existing identity system. The users must also use SAML 2.0 to pass authentication and authorization information to AWS. This API operation is useful in organizations that have integrated their identity systems (such as Windows AD or OpenLDAP) with software that can produce SAML assertions. Such an integration provides information about user identity and permissions (such as Active Directory Federation Services or Shibboleth).

GetFederationToken—Federation Through a Custom Identity Broker

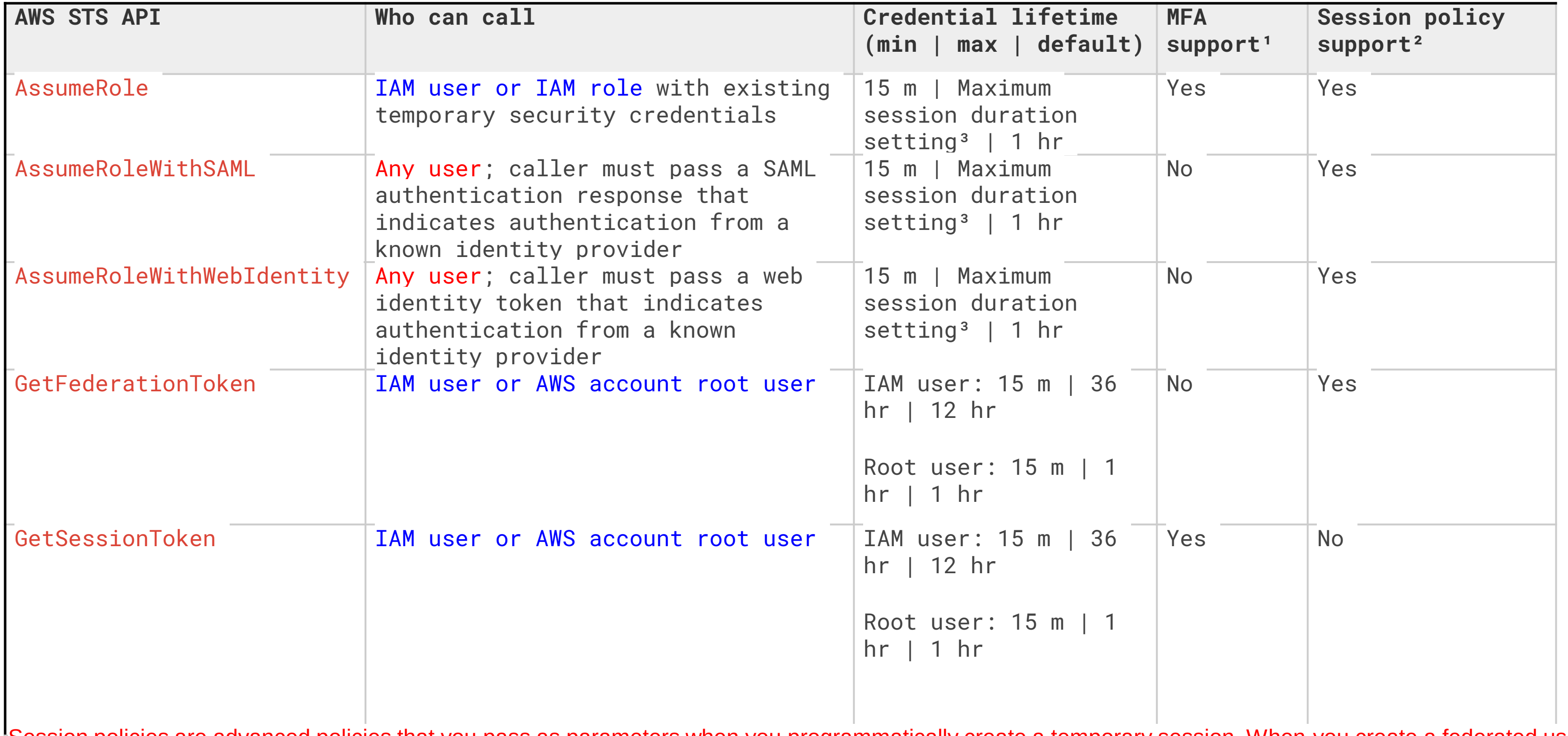
The GetFederationToken API operation returns a set of temporary security credentials for federated users. This API differs from AssumeRole in that the default expiration period is substantially longer (12 hours instead of one hour). Additionally, you can use the DurationSeconds parameter to specify a duration for the temporary security credentials to remain valid. The resulting credentials are valid for the specified duration, between 900 seconds (15 minutes) to 129,600 seconds (36 hours).The longer expiration period can help reduce the number of calls to AWS because you do not need to get new credentials as often. When you make a request to get temporary security credentials for a federated user, you use the credentials of a specific user identity (an IAM user) to make the request.

GetSessionToken—Temporary Credentials for Users in Untrusted Environments

The GetSessionToken API operation returns a set of temporary security credentials to an existing IAM user. This is useful for providing enhanced security, such as allowing AWS requests only when MFA is enabled for the IAM user. Because the credentials are temporary, they provide enhanced security when you have an IAM user who accesses your resources through a less secure

Requesting Temporary Security Credentials

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_request.html



AWS STS API	Who can call	Credential lifetime (min | max | default)	MFA support[1]	Session policy support[2]
AssumeRole	IAM user or IAM role with existing temporary security credentials	15 m | Maximum session duration setting[3] | 1 hr	Yes	Yes
AssumeRoleWithSAML	Any user; caller must pass a SAML authentication response that indicates authentication from a known identity provider	15 m | Maximum session duration setting[3] | 1 hr	No	Yes
AssumeRoleWithWebIdentity	Any user; caller must pass a web identity token that indicates authentication from a known identity provider	15 m | Maximum session duration setting[3] | 1 hr	No	Yes
GetFederationToken	IAM user or AWS account root user	IAM user: 15 m | 36 hr | 12 hr Root user: 15 m | 1 hr | 1 hr	No	Yes
GetSessionToken	IAM user or AWS account root user	IAM user: 15 m | 36 hr | 12 hr Root user: 15 m | 1 hr | 1 hr	Yes	No

Session policies are advanced policies that you pass as parameters when you programmatically create a temporary session. When you create a federated user session and pass session policies, the resulting session's permissions are the intersection of the IAM user's identity-based policy and the session policies. You cannot use the session policy to grant more permissions than those allowed by the identity-based policy of the user that is being federated.

Providing Access to AWS Accounts Owned by Third Parties

When third parties require access to your organization's AWS resources, you can use roles to delegate access to them. For example, a third party might provide a service for managing your AWS resources. With IAM roles, you can grant these third parties access to your AWS resources without sharing your AWS security credentials. Instead, the third party can access your AWS resources by assuming a role that you create in your AWS account.

Third parties must provide you with the following information for you to create a role that they can assume:

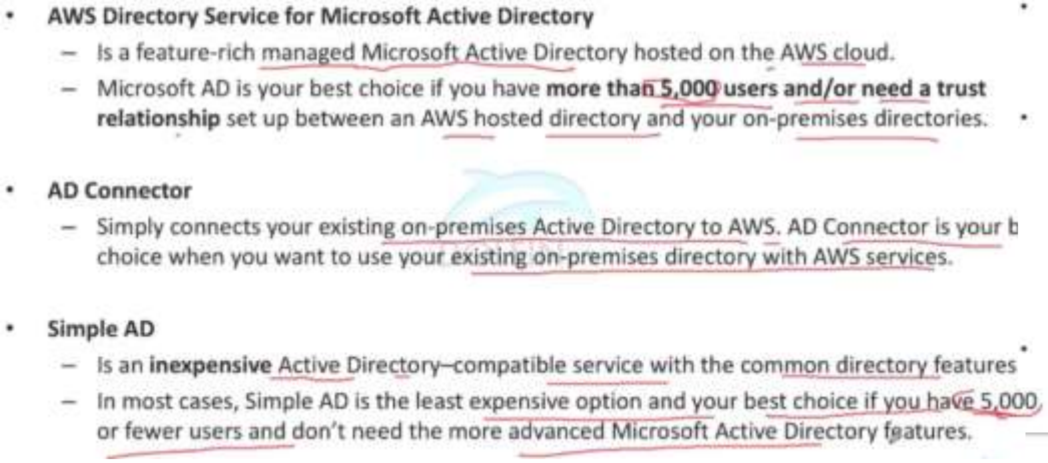
- **The third party's AWS account ID**. You specify their AWS account ID as the principal when you define the trust policy for the role.
- **An external ID** to uniquely associate with the role. The external ID can be any secret identifier that is known by you and the third party. For example, you can use an invoice ID between you and the third party, but do not use something that can be guessed, like the name or phone number of the third party. You must specify this ID when you define the trust policy for the role. The third party must provide this ID when they assume the role. For more information about the external ID, see How to Use an External ID When Granting Access to Your AWS Resources to a Third Party.
- **The permissions** that the third party requires to work with your AWS resources. You must specify these permissions when defining the role's permission policy. This policy defines what actions they can take and what resources they can access.

After you create the role, you must **provide the role's Amazon Resource Name (ARN) to the third party**. They require your role's ARN in order to assume the role.

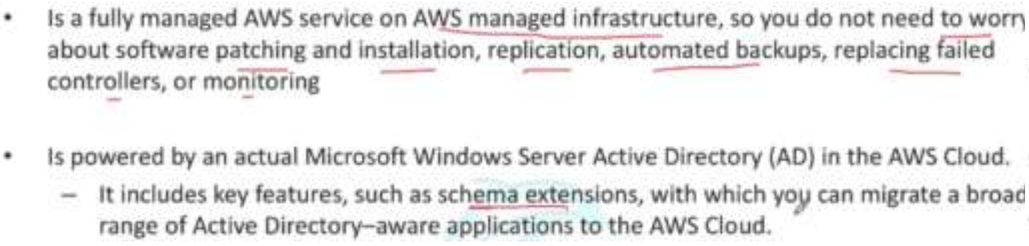
AWS Directory Service

(AWS MS AD)

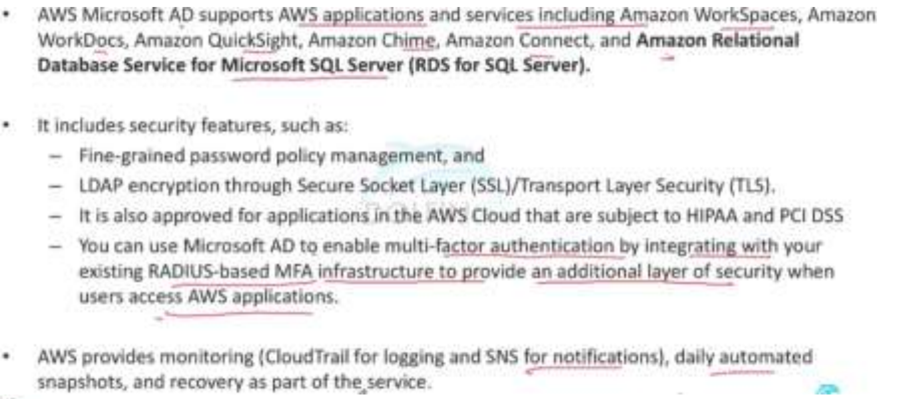
AWS MS AD - Intro



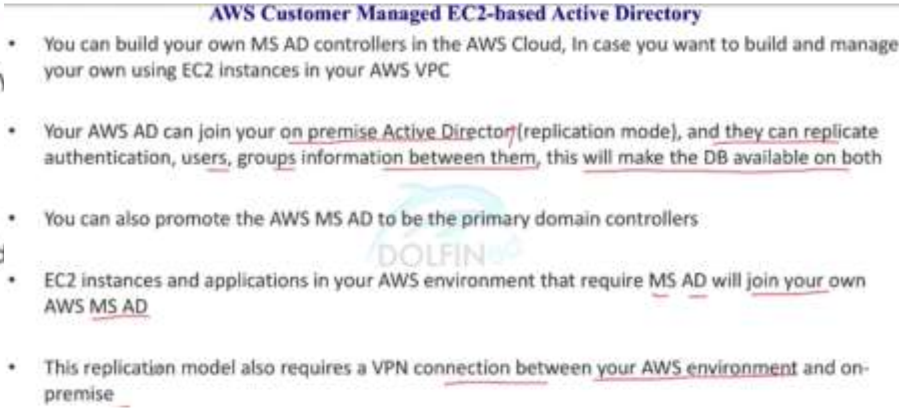
- **AWS Directory Service for Microsoft Active Directory**
  - Is a feature-rich managed Microsoft Active Directory hosted on the AWS cloud.
  - Microsoft AD is your best choice if you have **more than 5,000 users and/or need a trust relationship** set up between an AWS hosted directory and your on-premises directories.
- **AD Connector**
  - Simply connects your existing on-premises Active Directory to AWS. AD Connector is your b choice when you want to use your existing on-premises directory with AWS services.
- **Simple AD**
  - Is an **inexpensive** Active Directory–compatible service with the common directory features
  - In most cases, Simple AD is the least expensive option and your best choice if you have 5,000 or fewer users and don't need the more advanced Microsoft Active Directory features.



- Is a fully managed AWS service on AWS managed infrastructure, so you do not need to worr about software patching and installation, replication, automated backups, replacing failed controllers, or monitoring
- Is powered by an actual Microsoft Windows Server Active Directory (AD) in the AWS Cloud.
  - It includes key features, such as schema extensions, with which you can migrate a broad range of Active Directory–aware applications to the AWS Cloud.



- AWS Microsoft AD supports AWS applications and services including Amazon WorkSpaces, Amazon WorkDocs, Amazon QuickSight, Amazon Chime, Amazon Connect, and **Amazon Relational Database Service for Microsoft SQL Server (RDS for SQL Server).**
- It includes security features, such as:
  - Fine-grained password policy management, and
  - LDAP encryption through Secure Socket Layer (SSL)/Transport Layer Security (TLS).
  - It is also approved for applications in the AWS Cloud that are subject to HIPAA and PCI DSS
  - You can use Microsoft AD to enable multi-factor authentication by integrating with your existing RADIUS-based MFA infrastructure to provide an additional layer of security when users access AWS applications.
- AWS provides monitoring (CloudTrail for logging and SNS for notifications), daily automated snapshots, and recovery as part of the service.



AWS Customer Managed EC2-based Active Directory

- You can build your own MS AD controllers in the AWS Cloud, In case you want to build and manage your own using EC2 instances in your AWS VPC
- Your AWS AD can join your on premise Active Director (replication mode), and they can replicate authentication, users, groups information between them, this will make the DB available on both
- You can also promote the AWS MS AD to be the primary domain controllers
- EC2 instances and applications in your AWS environment that require MS AD will join your own AWS MS AD
- This replication model also requires a VPN connection between your AWS environment and on-premise

MSAD does not support replication mode when connecting to your on-prem AD, only Trust relationship is supported

KMS

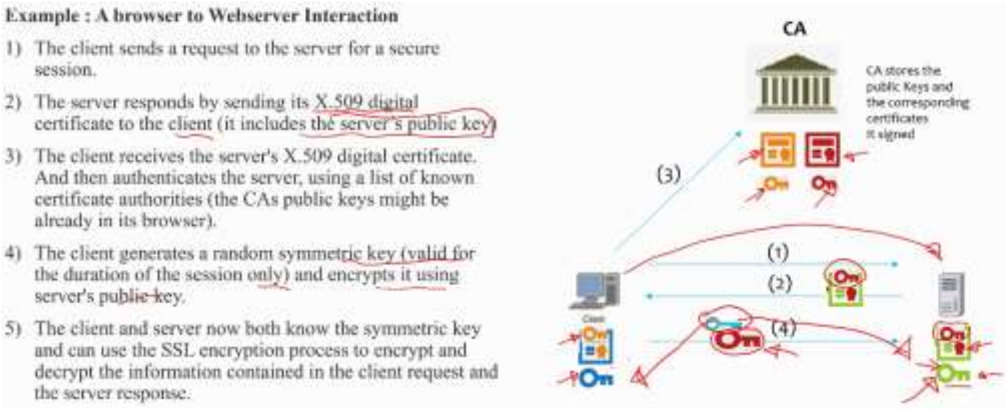
Cryptography- Concepts

2 types - Public key or Asymmetric cryptography (PKCS) and Symmetric key cryptography
Browser based applications interact with server with PKCS protocol
AWS KMS uses symmetric key encryption which is more efficient than PKE

To request a certificate:

1. Entity needs to submit a **Certificate Signing request (CSR)** to CA
2. The CSR itself is SIGNED using the entity’s private key
3. CA uses the entity’s public key to identify the entity and its ownership of the public key
4. If signature is validated, CA issues the certificate and signs it with it’s private key, binding the public key to the Distinguished name (Entity requesting the certificate)
5. All browsers are bundled with prominent CA’s public keys which enables it to validate server certificates

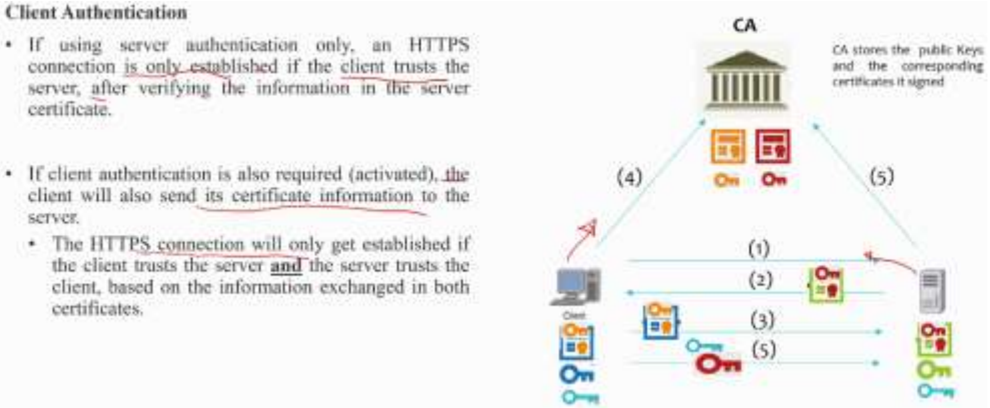
Certificate chains:
All major CA’s appoint 3rd party/resellers to sign certificates, in this case the entity’s SSL certificate must be signed with both the root and all intermediate CA’s. During certificate validation, the entire chain is validated from the SSL certificate → Intermediate CA → Root CA



Example : A browser to Webserver Interaction

1) The client sends a request to the server for a secure session.
2) The server responds by sending its X.509 digital certificate to the client (it includes the server's public key)
3) The client receives the server's X.509 digital certificate. And then authenticates the server, using a list of known certificate authorities (the CAs public keys might be already in its browser).
4) The client generates a random symmetric key (valid for the duration of the session only) and encrypts it using server's public key.
5) The client and server now both know the symmetric key and can use the SSL encryption process to encrypt and decrypt the information contained in the client request and the server response.

CA
CA stores the public Keys and the corresponding certificates it signed
(3)
(1)
(2)
(4)
Client



Client Authentication

- If using server authentication only, an HTTPS connection is only established if the client trusts the server, after verifying the information in the server certificate.
- If client authentication is also required (activated), the client will also send its certificate information to the server.
  - The HTTPS connection will only get established if the client trusts the server **and** the server trusts the client, based on the information exchanged in both certificates.

CA
CA stores the public Keys and the corresponding certificates it signed
(4)
(5)
(1)
(2)
(3)
(5)
Client

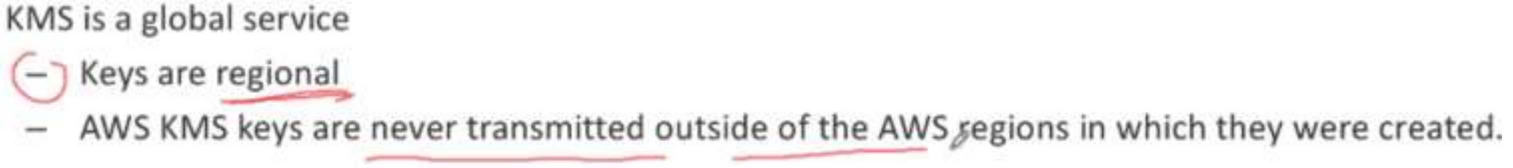
KMS - Concepts

A managed service that makes it easy for you to create and control the encryption keys used to encrypt your data. You can perform the following actions on your AWS KMS master keys:

- Create, describe, and list master keys
- Enable and disable master keys
- Create and view grants and access control policies for your master keys
- Enable and disable automatic rotation of the cryptographic material in a master key
- Import cryptographic material into an AWS KMS master key
- Tag your master keys for easier identification, categorizing, and tracking
- Create, delete, list, and update aliases, which are friendly names associated with your master keys
- Delete master keys to complete the key lifecycle

With AWS KMS you can also perform the following cryptographic functions using master keys:

- Encrypt, decrypt, and re-encrypt data
- Generate data encryption keys that you can export from the service in plaintext or encrypted under a master key that doesn't leave the service
- Generate random numbers suitable for cryptographic application



KMS is a global service

– Keys are regional

– AWS KMS keys are never transmitted outside of the AWS regions in which they were created.

Kms - Concepts

Customer Master Keys (CMKs):

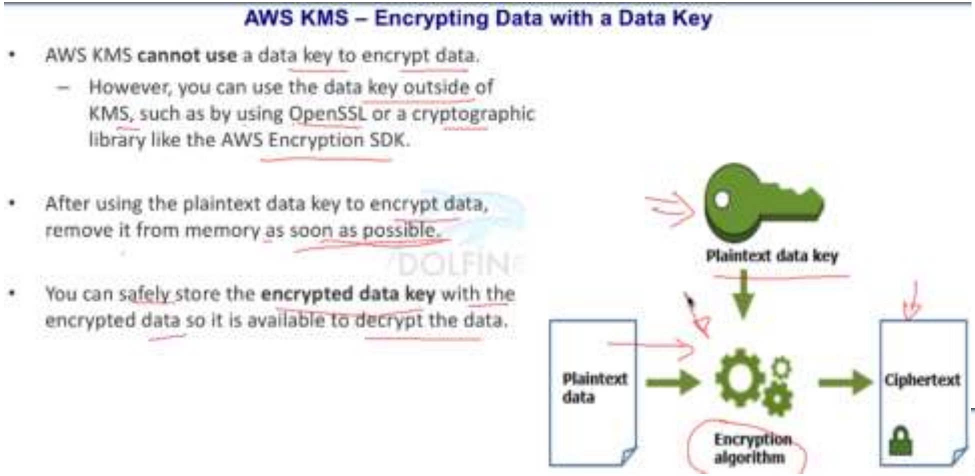
- Customer master keys are the primary resources in AWS KMS
- A customer master key (CMK) is a logical representation of a master key.
- The CMK includes metadata, such as the key ID, creation date, description, and key state.
- The CMK also contains the key material used to encrypt and decrypt data.
- You can use a CMK to encrypt and decrypt up to 4 KB (4096 bytes) of data.
- Typically, you use CMKs to generate, encrypt, and decrypt the data keys that you use outside of AWS KMS to encrypt your data. This strategy is known as **envelope encryption**.
- CMKs are created in AWS KMS and never leave AWS KMS unencrypted.
- Some services use CMK others use AWSMgK others give a choice, e.g. EBS provides a choice
- A **default Master CMK** specific to each service is created in your account as a convenience the first time you try to create and encrypted resource
- Alternatively you can create a custom master key to use in your own appl or from a aws service

Type of CMK	**Can view CMK metadata**	**Can manage CMK**	**Used only for my AWS account**
Customer managed CMK	Yes	Yes	Yes
AWS managed CMK **(used by AWS services automatically such as S3, EBS etc. identifiable by aws/s3, aws/redshift etc)**	Yes	No	Yes
AWS owned CMK	No	No	No

Data Keys:

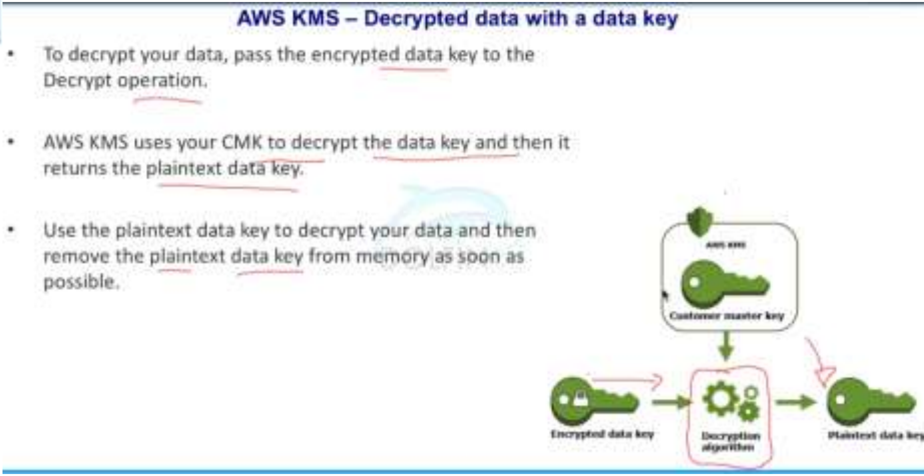
Data keys are encryption keys that you can use to encrypt data, including large amounts of data and other data encryption keys. You can use CMKs to generate, encrypt, and decrypt data keys. However, KMS does not store, manage, or track your data keys, or perform cryptographic operations with data keys. You must use and manage data keys outside of KMS.

Kms - Using Data Key



AWS KMS – Encrypting Data with a Data Key

- AWS KMS **cannot use** a data key to encrypt data.
  - However, you can use the data key outside of KMS, such as by using OpenSSL or a cryptographic library like the AWS Encryption SDK.
- After using the plaintext data key to encrypt data, remove it from memory as soon as possible.
- You can safely store the **encrypted data key** with the encrypted data so it is available to decrypt the data.



AWS KMS – Decrypted data with a data key

- To decrypt your data, pass the encrypted data key to the Decrypt operation.
- AWS KMS uses your CMK to decrypt the data key and then it returns the plaintext data key.
- Use the plaintext data key to decrypt your data and then remove the plaintext data key from memory as soon as possible.

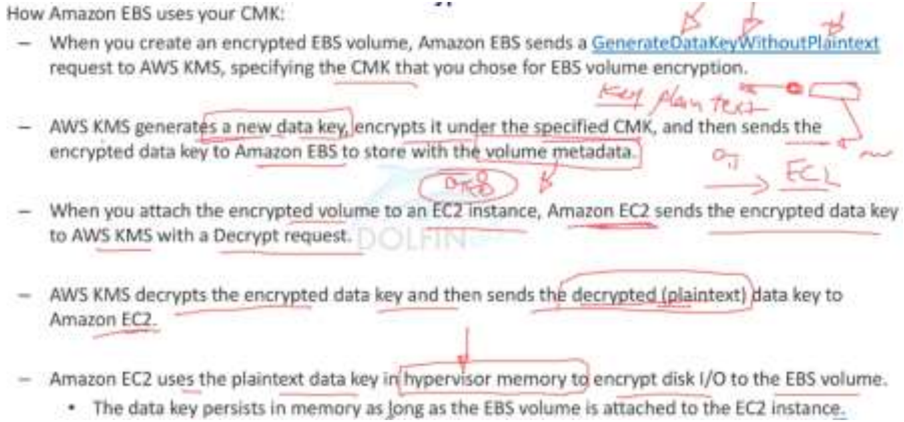
How EBS Uses KMS

EBS encryption

- ★When you attach an encrypted EBS volume to a supported EC2 instance type, data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted.
- ★The encryption occurs on the servers that host Amazon EC2 instances.
- ★This feature is supported on all Amazon EBS volume types.
- ★You access encrypted volumes the same way you access other volumes; encryption and decryption are handled transparently and they require no additional action from you, your EC2 instance, or your application.
- ★Snapshots of encrypted volumes are automatically encrypted, and volumes that are created from encrypted snapshots are also automatically encrypted.
- ★The encryption status of an EBS volume is determined when you create the volume. You cannot change the encryption status of an existing volume. However, you can migrate data between encrypted and unencrypted volumes and apply a new encryption status while copying a snapshot.

Using CMKs and Data Keys:

When you create an encrypted EBS volume, you specify an KMS customer master key (CMK). By default, EBS uses the AWS managed CMK for Amazon EBS in your account. However, you can specify a customer managed CMK. EBS uses the CMK that you specify to generate a unique data key for each volume. It stores an encrypted copy of the data key with the volume. Then, when you attach the volume to an Amazon EC2 instance, EBS uses the data key to encrypt all disk I/O to the volume.



How Amazon EBS uses your CMK:

- When you create an encrypted EBS volume, Amazon EBS sends a GenerateDataKeyWithoutPlaintext request to AWS KMS, specifying the CMK that you chose for EBS volume encryption.

Key Plain text

- AWS KMS generates a new data key, encrypts it under the specified CMK, and then sends the encrypted data key to Amazon EBS to store with the volume metadata.

DKB

EC2

- When you attach the encrypted volume to an EC2 instance, Amazon EC2 sends the encrypted data key to AWS KMS with a Decrypt request.
- AWS KMS decrypts the encrypted data key and then sends the decrypted (plaintext) data key to Amazon EC2.
- Amazon EC2 uses the plaintext data key in hypervisor memory to encrypt disk I/O to the EBS volume.
  - The data key persists in memory as long as the EBS volume is attached to the EC2 instance.

CloudHSM

CloudHSM - Concepts

AWS CloudHSM provides hardware security modules in the AWS Cloud. A hardware security module (HSM) is a computing device that processes cryptographic operations and provides secure storage for cryptographic keys. When you use an HSM from AWS CloudHSM, you can perform a variety of cryptographic tasks:

- ❏ Generate, store, import, export, and manage cryptographic keys, including symmetric keys and asymmetric key pairs.
- ❏ Use symmetric and asymmetric algorithms to encrypt and decrypt data.
- ❏ Use cryptographic hash functions to compute message digests and (HMACs).
- ❏ Cryptographically sign data (including code signing) and verify signatures.
- ❏ Generate cryptographically secure random data.

If you want a managed service for creating and controlling your encryption keys, but you don't want or need to operate your own HSM, consider using AWS Key Management Service.

CloudHSM - Use Cases

Offload the SSL/TLS Processing for Web Servers:

Web servers and their clients (web browsers) can use SSL or TLS. The web server uses a public–private key pair and an SSL/TLS public key certificate to establish an HTTPS session with each client. This process involves a lot of computation for the web server, but you can offload some of this to the HSMs in your AWS CloudHSM cluster. This is sometimes known as SSL acceleration. Offloading reduces the computational burden on your web server and provides extra security by storing the server's private key in the HSMs.

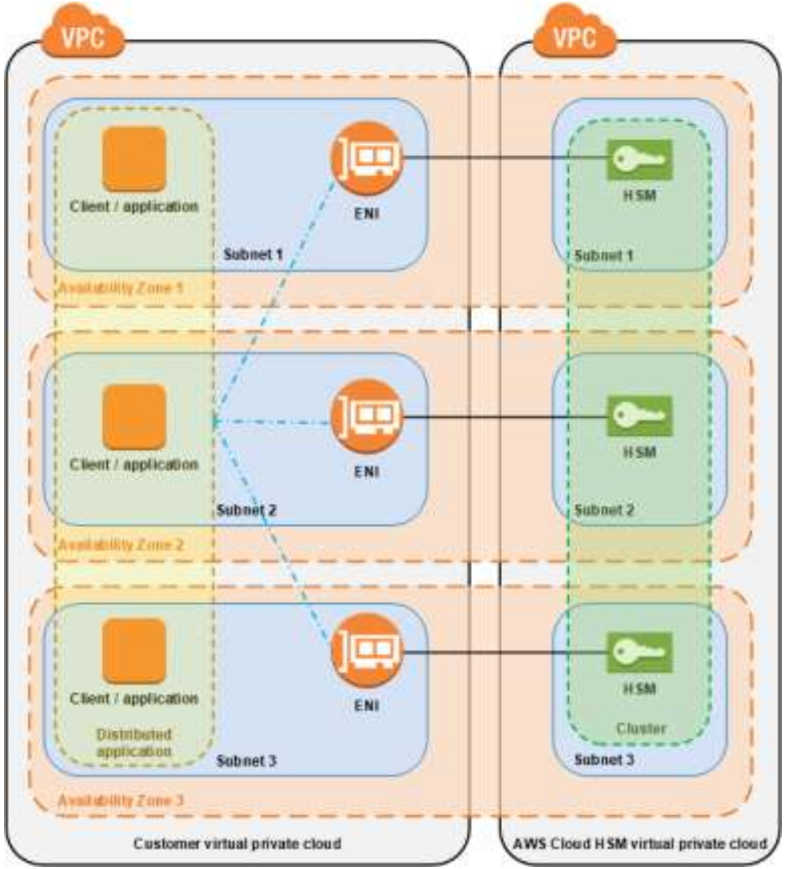
Protect the Private Keys for an Issuing Certificate Authority (CA):

In a public key infrastructure (PKI), a certificate authority (CA) is a trusted entity that issues digital certificates. These digital certificates bind a public key to an identity (a person or organization) by means of public key cryptography and digital signatures. To operate a CA, you must maintain trust by protecting the private key that signs the certificates issued by your CA. You can store the private key in the HSM in your AWS CloudHSM cluster, and use the HSM to perform the cryptographic signing operations.

Enable Transparent Data Encryption (TDE) for Oracle Databases:

Some versions of Oracle's database software offer a feature called Transparent Data Encryption (TDE). With TDE, the database software encrypts data before storing it on disk. The data in the database's table columns or tablespaces is encrypted with a table key or tablespace key. These keys are encrypted with the TDE master encryption key. You can store the TDE master encryption key in the HSMs in your AWS CloudHSM cluster, which provides additional security.

CloudHSM - Cluster Architecture



Each time you create an HSM, you specify the cluster and Availability Zone for the HSM. By putting the HSMs in different Availability Zones, you achieve redundancy and high availability in case one Availability Zone is unavailable.

When you create an HSM, AWS CloudHSM puts an elastic network interface (ENI) in the specified subnet in your AWS account. The elastic network interface is the interface for interacting with the HSM.

The HSM resides in a separate VPC in an AWS account that is owned by AWS CloudHSM. The HSM and its corresponding network interface are in the same Availability Zone.

To interact with the HSMs in a cluster, you need the AWS CloudHSM client software. Typically you install the client on Amazon EC2 instances, known as client instances, that reside in the same VPC as the HSM ENIs, as shown in the following figure. That's not technically required though; you can install the client on any compatible computer, as long as it can connect to the HSM ENIs. The client communicates with the individual HSMs in your cluster through their ENIs.

CloudHSM - HSM Users

Precrypto Officer (PRECO)

The PRECO is a temporary user that exists only on the first HSM in an AWS CloudHSM cluster. Used for activiating the cluster, the PRECO user becomes a crypto officer (CO).

Crypto Officer (CO)

A CO can perform user mgmt operations. eg. a CO can create and delete users and change user passwords.

Crypto User (CU)

A CU can perform the following key management and cryptographic operations.

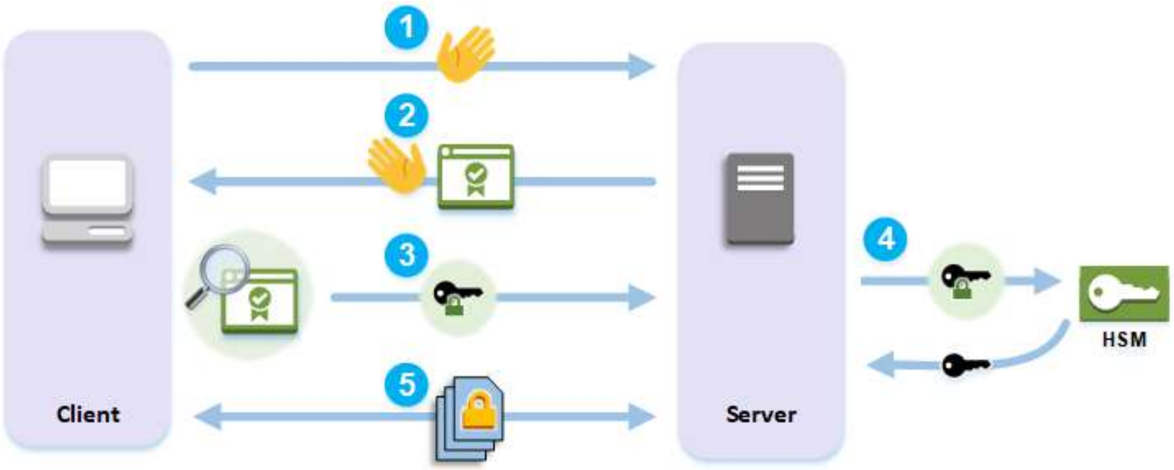
- **Key management** – Create, delete, share, import, and export cryptographic keys.
- **Cryptographic operations** – Use cryptographic keys for encryption, decryption, signing, verifying, and more.

Appliance User (AU)

The AU can perform cloning and synchronization operations. AWS CloudHSM uses the AU to synchronize the HSMs in an AWS CloudHSM cluster. The AU exists on all HSMs provided by AWS CloudHSM, and has limited permissions.

	Crypto Officer (CO)	Crypto User (CU)	Appliance User (AU)	Unauthenticated User
Get basic cluster info[1]	Yes	Yes	Yes	Yes
Zeroize an HSM[2]	Yes	Yes	Yes	Yes
Change own password	Yes	Yes	Yes	Not applicable
Change any user's password	Yes	No	No	No
Add, remove users	Yes	No	No	No
Get sync status[3]	Yes	Yes	Yes	No
Extract, insert masked objects[4]	Yes	Yes	Yes	No
Key management functions[5]	No	Yes	No	No
Encrypt, decrypt	No	Yes	No	No
Sign, verify	No	Yes	No	No
Generate digests and HMACs	No	Yes	No	No

Improve Your Web Server's Security with SSL/TLS Offload in AWS CloudHSM



1. The client sends a hello message to the server.
2. The server responds with a hello message and sends the server's certificate.
3. The client performs the following actions:
   a. Verifies that the SSL/TLS server certificate is signed by a root certificate that the client trusts.
   b. Extracts the public key from the server certificate.
   c. Generates a premaster secret and encrypts it with the server's public key.
   d. Sends the encrypted premaster secret to the server.
4. To decrypt the client's premaster secret, the server sends it to the HSM. The HSM uses the private key in the HSM to decrypt the premaster secret and then it sends the premaster secret to the server. Independently, the client and server each use the premaster secret and some information from the hello messages to calculate a master secret.
5. The handshake process ends. For the rest of the session, all messages sent between the client and the server are encrypted with derivatives of the master secret.

WAF

AWS WAF can be applied to APIG, ALB or CF

At the simplest level, AWS WAF lets you choose one of the following behaviors:

- **Allow all requests except the ones that you specify**
- **Block all requests except the ones that you specify**
- **Count the requests that match the properties that you specify**

Additionally:

- Define conditions by using characteristics of web requests such as the following:
  - IP addresses that requests originate from, Country that requests originate from, Values in request headers, Strings that appear in requests, either specific strings or string that match regular expression (regex) patterns, Length of requests, Presence of SQL code that is likely to be malicious (known as *SQL injection*), XSS
- Rules that can allow, block, or count web requests that meet the specified conditions. Alternatively, rules can block or count web requests that not only meet the specified conditions, but also exceed a specified number of requests in any 5-minute period.
- Rules that you can reuse for multiple web applications.
- Real-time metrics and sampled web requests.
- Automated administration using the AWS WAF API

AWS Shield

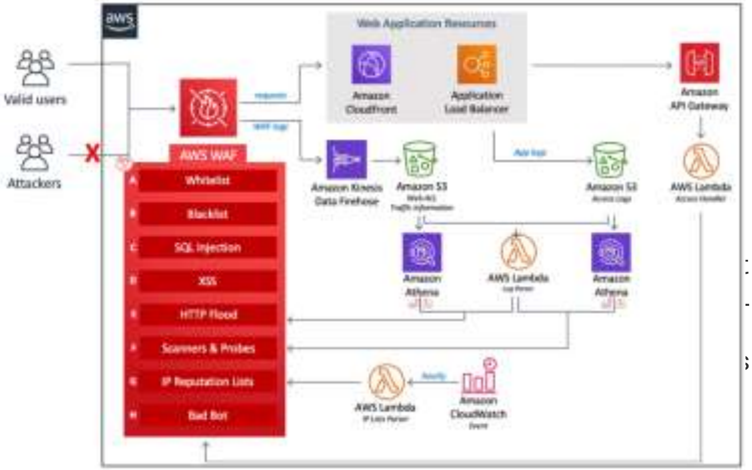
You can use AWS WAF web access control lists (web ACLs) to help minimize the effects of a DDoS attack. For additional protection against DDoS attacks, AWS provides AWS Shield Standard & Advanced. AWS Shield Advanced provides expanded DDoS attack protection for your Amazon EC2 instances, Elastic Load Balancing load balancers, CloudFront distributions, and Route 53 hosted zones. AWS Shield Advanced incurs additional charges.

AWS WAF CAN BE APPLIED TO APIG, ALB OR CF

Types of implementations:

1. **Custom - WAF Security Automation**

AWS offers a solution that uses AWS CloudFormation to automat
filter common web-based attacks. Users can select from precon
included in an AWS WAF web access control list (web ACL). Onc
inspecting web requests to the user's existing Amazon CloudFr
and block
them when applicable.



to
es
s,

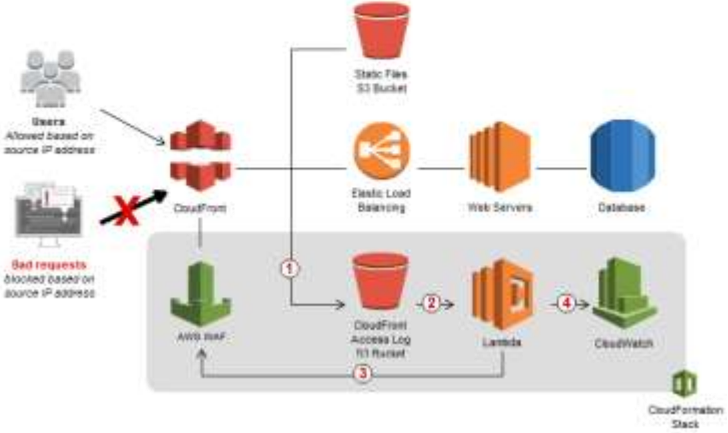
2. **Custom - Blocking IP addresses that submit bad requests**

Using AWS Lambda, you can set a threshold of how many
bad requ
tolerate
from a given IP address. A bad request is one for which
CloudFront origin re

400, Bad Request 403, Forbidden 404, Not Found 405, Meth

If users (based on IP addresses) exceed this error code threshold,
Lambda automatically updates you

how long requests from those IP addresses
should be blocked.



3. **Managed rules for WAF**

Pre-configured rules from 3rd party security sellers. Can be applied in addition to custom rules

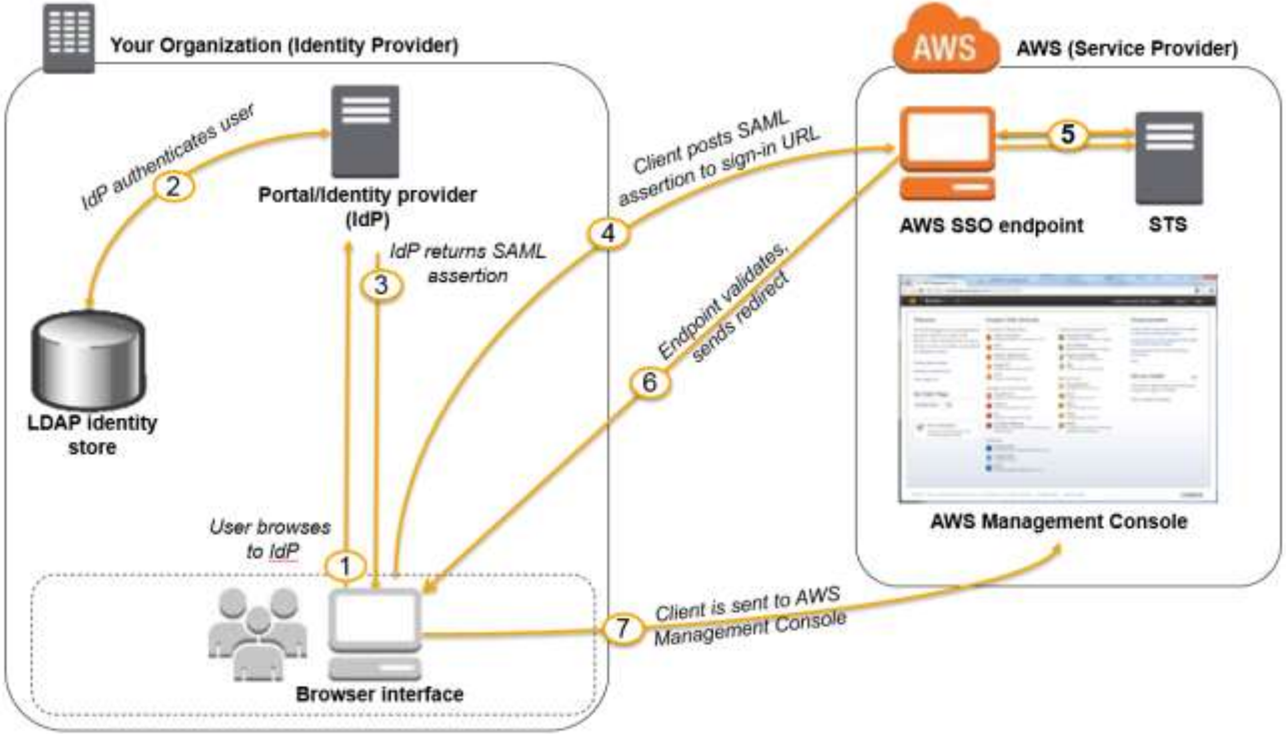
Difference b/w Shield Standard and Shield Advanced

https://aws.amazon.com/shield/getting-started

Feature (only differences)	Shield Standard	Shield Advanced
Monitoring - Application traffic monitoring	No	Yes
Mitigations - Additional DDoS mitigation capacity for large attacks	No	Yes
Mitigations - DRT-driven application layer (Layer 7) mitigations	No	Yes with DRT response team
Visibility - Layer 3/Layer 4 attack notification **Visibility** - Layer 7 attack notification **Visibility** - Layer 3/Layer 4/ Layer 7 attack historical report	No	Yes
DRT Support - Custom mitigations during attacks **DRT Support** - Post attack analysis	No	Yes
DDoS Cost Protection (Service credits for DDoS scaling charges) R53, CloudFront, ELB, EC2	No	Yes
Cost Monthly, Usage-based , SLA	No	Yes

IAM - Enabling SAML 2.0 Federated Users to Access the AWS Management Console

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_enable-console-saml.html



Your Organization (Identity Provider)
IdP authenticates user
2
Portal/Identity provider (IdP)
IdP returns SAML assertion
3
LDAP identity store
User browses to IdP
1
Browser interface
Client posts SAML assertion to sign-in URL
4
Endpoint validates, sends redirect
6
Client is sent to AWS Management Console
7
AWS
AWS (Service Provider)
5
AWS SSO endpoint
STS
AWS Management Console