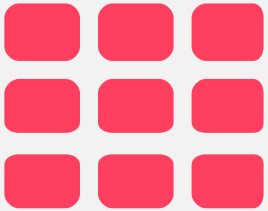

aws
certified
Solutions
Architect
Professional
DOLFINed



DESIGNING SECURE SOLUTIONS - AWS SECURITY, IDENTITY, AND COMPLIANCE SERVICES

AMAZON IDENTITY & ACCESS MANAGEMENT (IAM), STS, FEDEARATION, AND SINGLE SIGN-ON

Identity and Access Management

IAM Roles, IAM Service Roles & IAM Service-linked Roles

Review Topic : AWS IAM

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 and provided to the user.

- Roles can be assumed/used by any of the following:
  - An IAM user in the same, or different, AWS account as the role
  - A web service offered by AWS such as 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.com

Review Topic : AWS IAM

IAM Role and Resource Based Policies – The difference

- 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 that with an **IAM 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.

source: aws.amazon.co

Review Topic : AWS IAM

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.

source: aws.amazon.co

Review Topic : AWS IAM

IAM Role for EC2 instances

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

- You can specify a role for the instance at launch or after.
  - 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 can 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,

source: aws.amazon.com

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.

source: aws.amazon.co

Identity and Access Management

- IAM Role Delegation
- IAM Roles for Cross-Account Access
- Resource-based policies for Cross Account Access

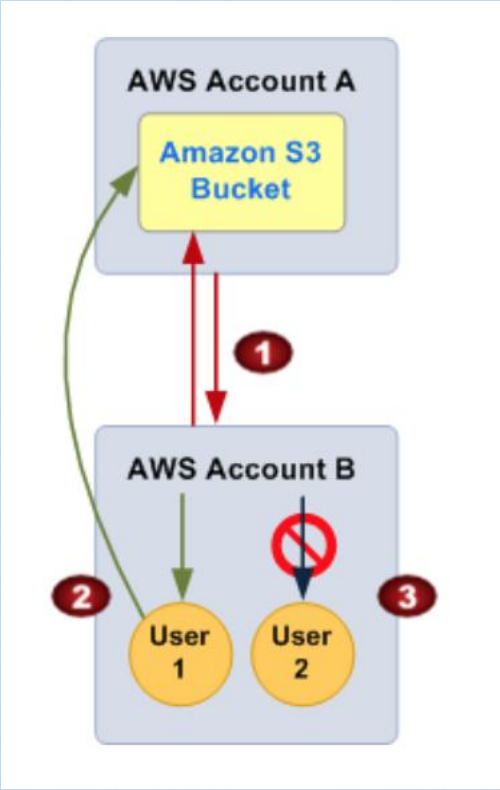
IAM Role Delegation

Delegation

- **Is** the granting of permission to someone to allow access to resources that you control.

- Delegation involves setting up a trust between the account that owns the resource **(the trusting account),** and the account that contains the users that need to access the resource **(the trusted account).**

- The trusted and trusting accounts can be any of the following:
  - The same account.
  - Two accounts that are both under your (organization's) control.
  - Two accounts owned by different organizations.



source: aws.amazon.co

IAM Role Delegation

- To delegate permission to access a resource, you create **an IAM role that has two policies attached**.
  - The ***permissions policy*** (JSON format) where the actions and resources the role can use are defined.
    - It grants the user of the role the needed permissions to carry out the intended tasks on the resource.
  - The ***trust policy*** (JSON format) specifies which trusted accounts can grant its users permissions to assume the role.
    - It defines who can assume the role .
    - This **trusted entity** is included in the policy as the ***principal*** **element** in the document.

- When you **create a trust policy, you cannot specify a wildcard (*) as a principal**.
  - The trust policy on the role in the **trusting account** is one-half of the permissions.
  - The **other half is a permissions policy attached to the user in the trusted account** that allows that user to switch to or assume the role.

source: aws.amazon.com

Review Topic : AWS IAM

IAM Principal

- **Principal**
  - An entity in AWS that can perform actions and access resources.
  - A principal can be an AWS account root user, an IAM user, or a role.

- If you reference an AWS account as principal, it generally means any principal defined within that account.

source: aws.amazon.co

Review Topic : AWS IAM

IAM Role for Cross-Account Access

- 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.
  - Instead, use IAM roles.
  - You can define a role in the trusting account, that specifies what permissions the IAM users in the other, trusted, account are allowed.
  - You can also designate which **AWS accounts** have the IAM users that can assume the role.
    - AWS allows you to define AWS account in trust policies, Not IAM users
- Granting access to resources in one account to a trusted principal in a different account.
  - Roles are the primary way to grant cross-account access.

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

source: aws.amazon.com

Review Topic : AWS IAM

Cross Account Access using Resource Based Policies:

- 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.

- AWS services that support Resource-based Policies:
  - Amazon Simple Storage Service (S3) buckets,
  - Amazon Glacier vaults,
  - Amazon Simple Notification Service (SNS) topics, and
  - Amazon Simple Queue Service (SQS) queues.

source: aws.amazon.co

Identity and Access Management

Security Token Service (STS)

Review Topic : AWS STS

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**, **Role delegation**, **cross-account access**, and **IAM roles**.

- You can use the AWS Security Token Service (AWS STS) to create and provide **trusted users** with **temporary security credentials** that can control access to your AWS resources.
  - To request the temporary security credentials, use the AWS STS API Actions.

source: aws.amazon.co

STS – Temporary Credentials vs. to IAM Users' Credentials

- Temporary security credentials work almost identically to the long-term access key credentials that your IAM users can use, with the following differences:
  - Temporary security credentials **are *short-term***, as the name implies.
    - They can be configured to last for anywhere from a **few minutes to several hours**.
  - After the credentials expire, AWS no longer recognizes them or allows any kind of access from API requests made with them.
  - Temporary security credentials **are not stored with the user** but are generated dynamically and provided to the user when requested.
  - When (or even before) the temporary security credentials expire, the user can request new credentials, as long as the user requesting them still has permissions to do so.

source: aws.amazon.com

Review Topic : AWS STS

STS - Advantages

Advantages for using temporary credentials:

- You do not have to distribute or embed long-term AWS security credentials with an application.
- 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.

source: aws.amazon.com

Review Topic : AWS STS

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**
  - 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.amazon.com

Review Topic : AWS STS

STS API Actions

- **Passed policy support**.
  - You can pass an IAM policy as a parameter to most of the AWS STS APIs to be used in conjunction with other policies affecting the user (if any) to determine what the user is allowed to do with the temporary credentials that result from the API call.
  - This is not supported with GetSessionToken

- **MFA support**.
  - You can include information about a multi-factor authentication (MFA) device when you call the AssumeRole and GetSessionToken APIs.
  - This ensures that the temporary security credentials that result from the API call can be used only by users who are authenticated with an MFA device.

- There is a default expiration for the temporary security credentials that differs based on which STS API the call was made to (AssumeRole, AssumeRoleWithSAML, AssumeRoleWithWebIdentity, default is 1 hour, and with GetSessionToken, GetFederationToken default is 12 hours)
  - You can also send the requested duration of the Token, and this can be from 15 minutes all the way to few hours (Depends on which STS API you call, up to 36 hours)

source: aws.amazon.com

Identity and Access Management

- Web Identity Federation
- Federation using SAML 2.0
- Single Sign-on Using SAML 2.0 or a Custom IdP
- IAM best practices

AssumeRoleWithWebIdentity STS API

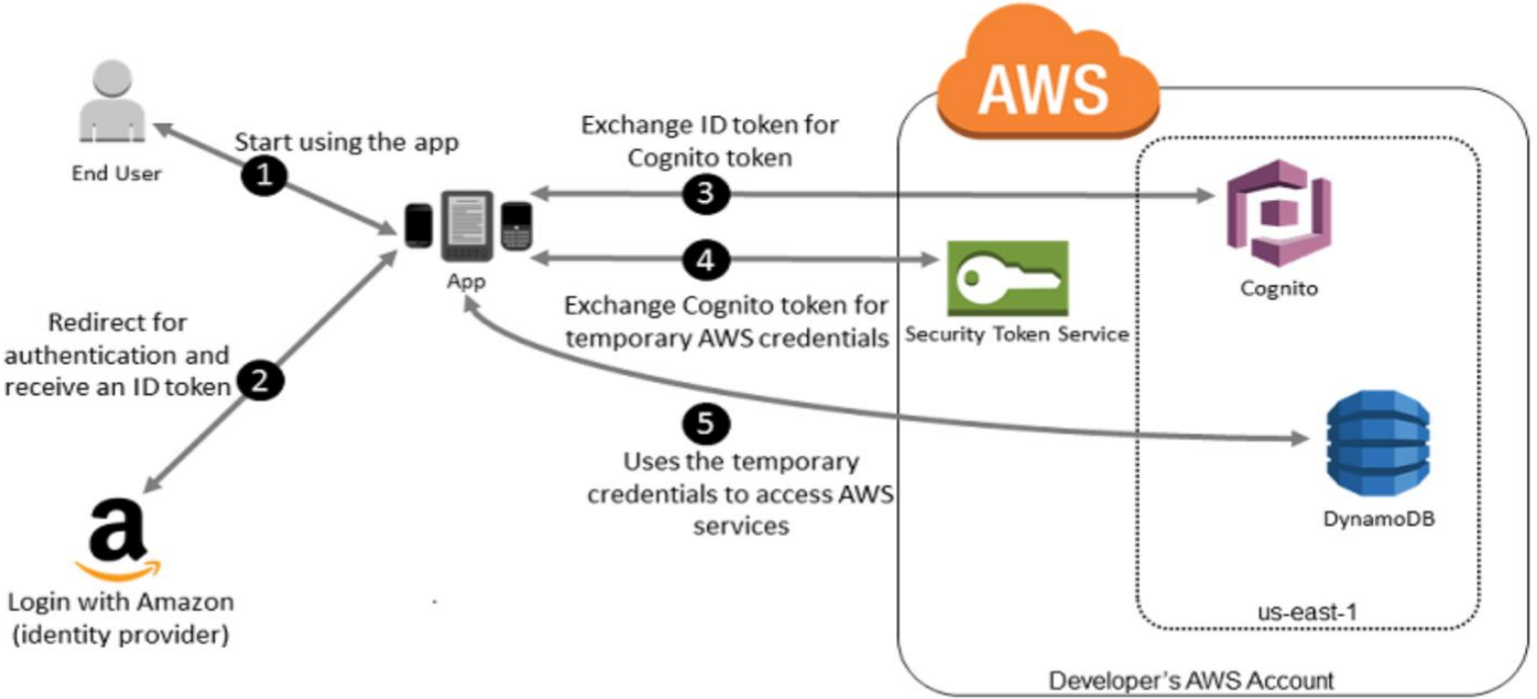
AssumeRoleWithWebIdentity

- Who can call: Any user; caller must pass a web identity token that indicates authentication from a known identity provider
- The **permission policy of the role that is being assumed determines the permissions** for the temporary security credentials returned by and AssumeRoleWithWebIdentity.
  - You define these permissions when you create or update the role.
- **Optionally,** you can pass a separate policy as a parameter of the AssumeRoleWithWebIdentity API call.

source: aws.amazon.com

Identity Federation

Web Identity Federation through IdP – How it works – Mobile App example



source: aws.amazon.com

SAML 2.0 – Identity Federation

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 into all your SAML-enabled applications by using a single set of credentials.

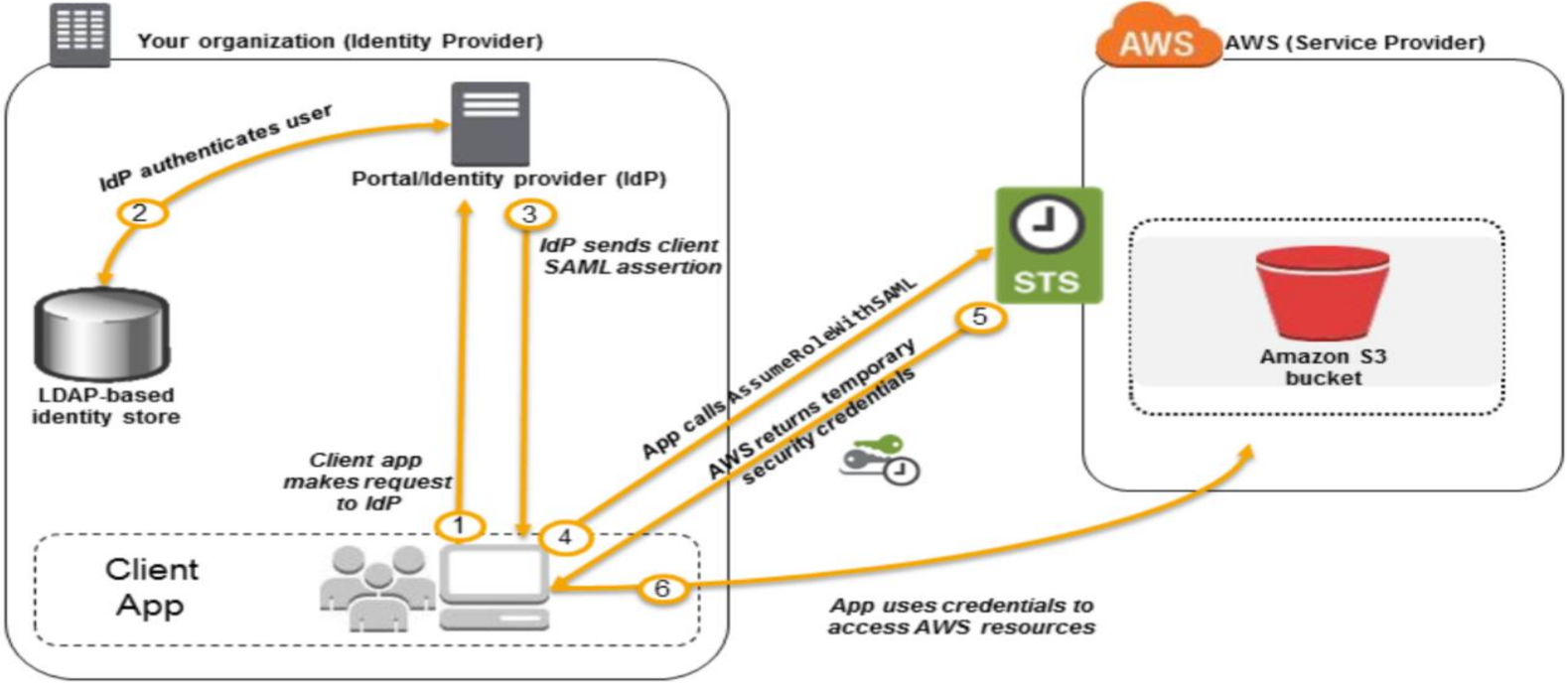
- By enabling SAML authentication, you also can manage access to your applications centrally.

- 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.

source: aws.amazon.co

Review Topic : Active Directory Services

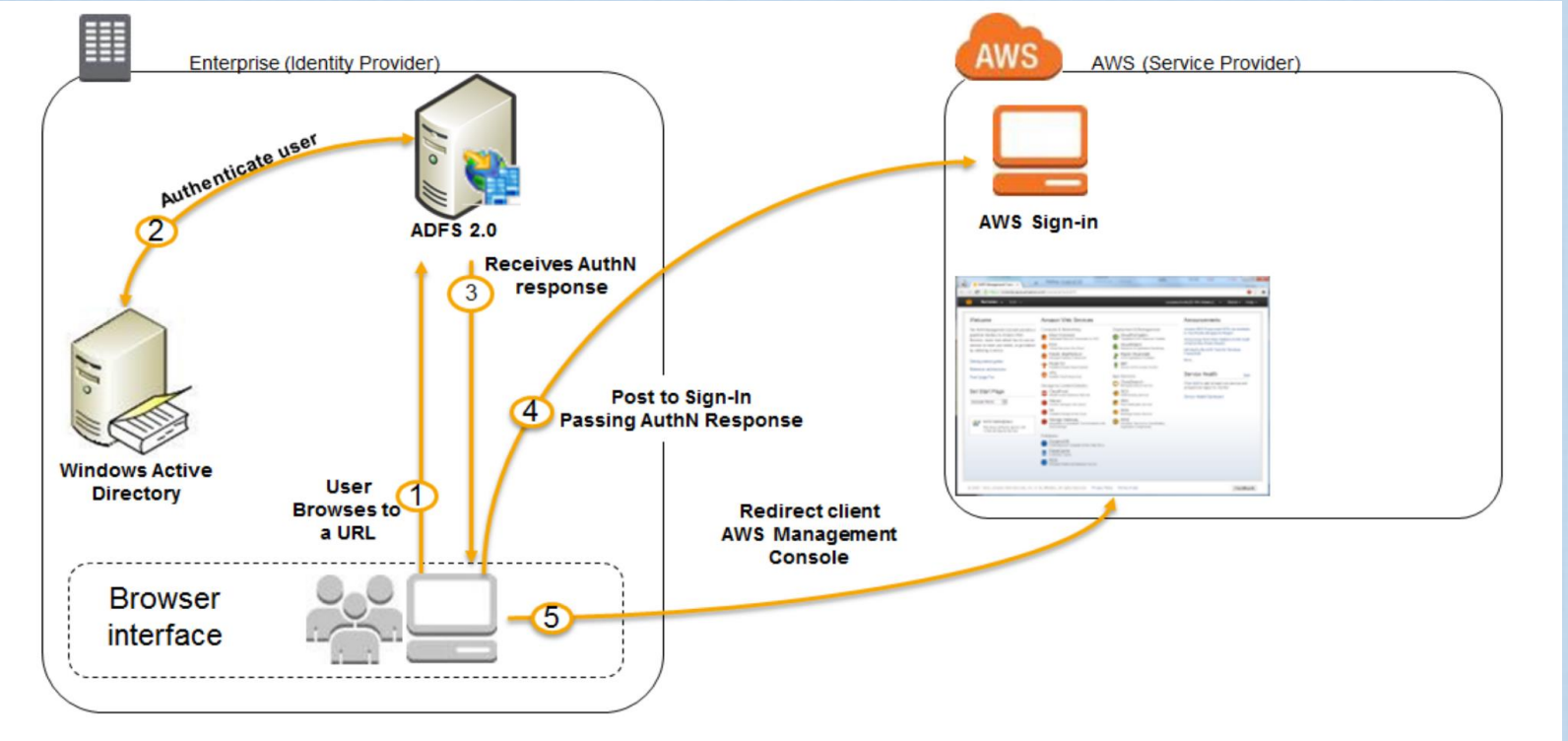
SAML 2.0 - Identity Federation



source: aws.amazon.co

Review Topic : Active Directory Services

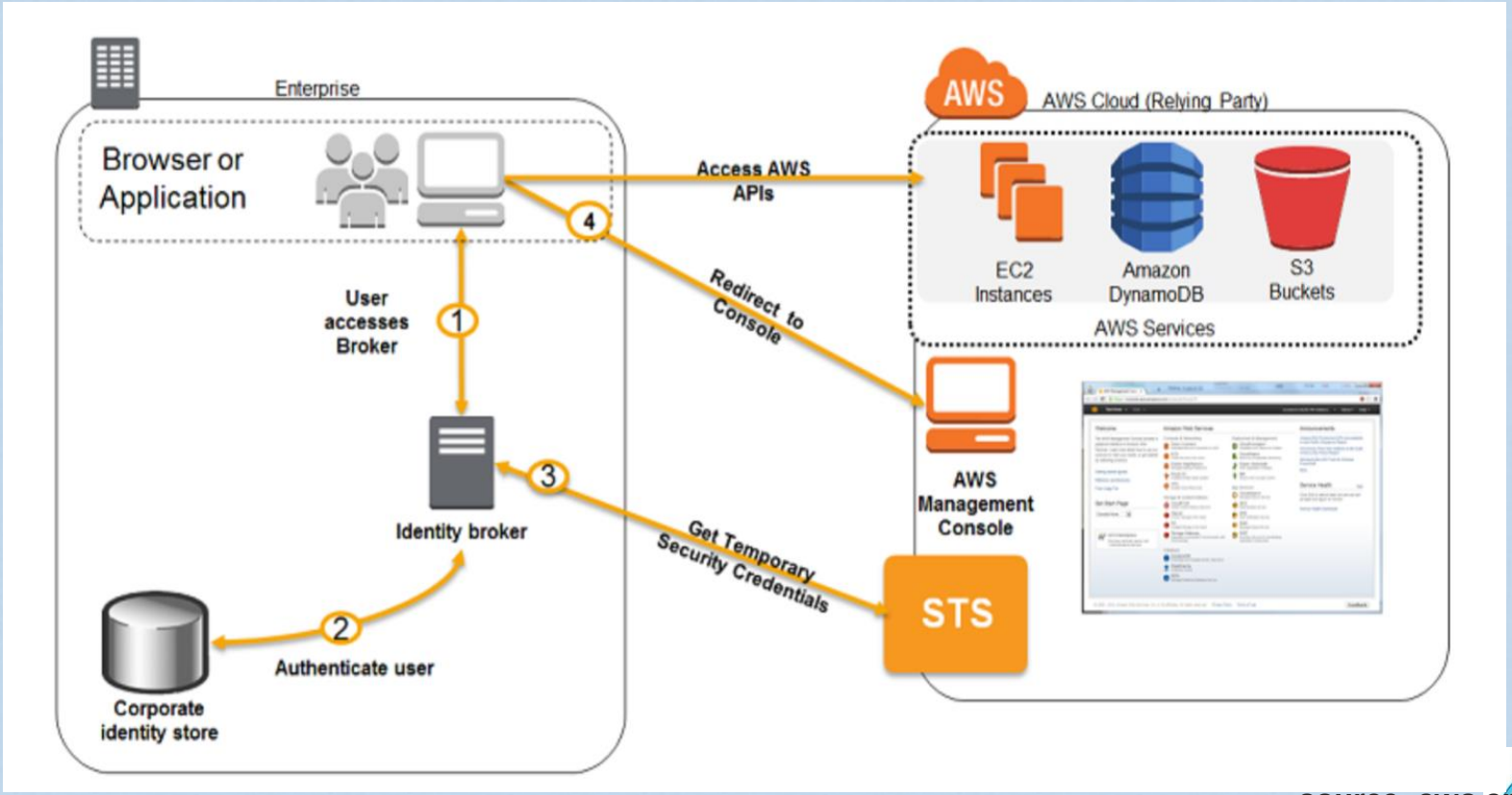
SAML 2.0 - Identity Federation



source: aws.amazon.co

Review Topic : STS

Federation – Using GetFederationToken through Identity Broker/Proxy



source: aws.amazon.co

Review Topic : AWS IAM

IAM Best Practices

- Lock away your AWS account Root user access keys (an access key ID and secret access key)
- Create Individual IAM users
- Use AWS Defined Policies to Assign permissions whenever possible (AWS ready policies)
- Use Groups to assign permissions to IAM users
- Grant Least Privilege
- Use Access levels to Review IAM permissions (AWS categorizes each service action into one of four *access levels* based on what each action does: List, Read, Write, or Permissions management.)
- Co
- Enable MFA for Privileged Users
- Use Roles for Applications that run on AWS EC2 instances
- Delegate by using Roles instead of sharing Credentials
- Rotate Credentials Regularly
- Remove Unnecessary Credentials
- Use Policy Conditions for Extra Security
- Monitor Activity in Your AWS Account
- nfigure a Strong Password policy for Users

source: aws.amazon.co

AWS DIRECTORY SERVICES

AWS Active Directory Services

Option	Fully Managed by AWS	SSO support	RDS MS SQL support	MFA Support	Best Use case	Snapshots for Backup and DR	Same Policies on premise and in AWS	Can authenticate to AWS console w/out SAML2.0	Cloud Trail and SNS integration	Supports Multi AZ
AWS Active Directory Service for MS AD	Yes - Standard < 5000 users, and Enterprise >5000 users	Yes, SAML 2.0 , No replication option to On Premise AD (VPN)	Yes	Yes	> 5000 users, or when Trust with on-premise MS AD	Yes (Automated and Manual)	Can be achieved with Trust	Yes	Yes	Yes (default)
AD Connector	Small <=500 users and Large <=5000 Users	Yes, however, No LDAP DB in the cloud, (Requires VPN or DX)	No	Yes	When you want to use your On-premise Existing AD directory with your AWS application	N/A	No policies in AWS, everything is on premise	Yes, allows users to log in to AWS using their AD credentials		N/A
Simple AD	Yes	Yes (Kerberos based), No trust relations or SAML based federation	No	No	<= 5000 users, Best for a low cost Active Directory compatible service in AWS	Yes (Automated and Manual)		Yes		Deployed on 2 EC2 instances in 2 different subnets in a VPC
You own MS AD on EC2 instances (unmanaged)	No , Small <=500 users and Large <=5000 Users	MS ADs in AWS can join On PremiseAD **(Replication) Not trust (VPN) But will need trust with AS for SSO**		Yes	Low scale, Low cost, basic features AD like requirement, and for LDAP aware applications	Not automated, you need to do your own snapshots of the EBS volumes	Yes	You need SAML2.0 for SSO	N/A	Your own Design

AWS CLOUDHSM

AWS Cloud HSM

- An HSM is a dedicated storage and data processing device that performs cryptographic operations using keys on the device.
  - An HSM typically provides tamper evidence, or resistance, to protect keys from unauthorized use.
  - A software-based authorization layer controls who can administer the HSM and which users or applications can use which keys in the HSM.

- AWS CloudHSM provides a FIPS 140-2 Level 3 validated single-tenant HSM cluster in your Amazon Virtual Private Cloud (VPC) to store and use your keys.
  - **FIPS** (Federal Information Processing Standards) are a set of standards that describe document processing, encryption algorithms and other information technology standards for use within non-military government agencies and by government contractors and vendors who work with the agencies.

AWS Cloud HSM

When you use AWS CloudHSM

- You have exclusive control over how your keys are used via an authentication mechanism independent from AWS.

- You interact with keys in your AWS CloudHSM cluster similar to the way you interact with your applications running in Amazon EC2.

- You can use AWS CloudHSM to support a variety of use cases, such as:
  - Digital Rights Management (DRM),
  - Public Key Infrastructure (PKI),
  - Document signing, and
  - Cryptographic functions.

AMAZON KEY MANAGEMENT SERVICE (KMS)

AWS Key Management Service

- Introduction
- AWS Managed Keys
- Customer Master Keys CMKs

AWS KMS

- KMS is a global service
  - Keys are regional
  - AWS KMS keys are never transmitted outside of the AWS regions in which they were created.
- AWS KMS now support both Symmetric and Asymmetric keys
- AWS KMS stores multiple copies of encrypted versions of your keys in systems that are designed for 99.999999999% durability to help assure you that your keys will be available when you need to access them.
- AWS KMS is deployed in multiple availability zones within an AWS region to provide high availability for your encryption keys.
- If you import keys into KMS, you must securely maintain a copy of your keys so that you can re-import them at any time.
- The master keys created on your behalf by AWS KMS or imported by you cannot be exported from the service.

AWS KMS – Customer Master Keys (CMKs)

- The primary resources in AWS KMS are Customer Master Keys (CMKs).
  - You can use a CMK to encrypt and decrypt **up to 4 kilobytes (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**
- AWS KMS stores, tracks, and protects your CMKs.
  - When you want to use a CMK, you access it through AWS KMS.
- AWS KMS helps you to protect your master keys by storing and managing them securely.
  - Master keys stored in AWS KMS, known as customer master keys (CMKs), never leave the AWS KMS FIPS validated hardware security modules unencrypted.
  - To use an AWS KMS CMK, you must call AWS KMS.
  - This strategy differs from data keys that AWS KMS returns to you, optionally in plaintext.

AWS KMS – Customer Master Keys (CMKs)

There are two types of CMKs in AWS accounts:

Customer managed CMKs

- These are CMKs that you create, manage, and use.
  - This includes enabling and disabling the CMK,
  - Rotating its cryptographic material, and
  - Establishing the IAM policies and key policies that govern access to the CMK,
  - As well as using the CMK in cryptographic operations.
- You can allow an AWS service to use a customer managed CMK on your behalf, but you retain control of the CMK.

AWS managed CMKs

- These are CMKs in your account that are created, managed, and used on your behalf by an AWS service that is integrated with AWS KMS.
- This CMK is unique to your AWS account **and region.**
- **Only the service that created the AWS managed CMK can use it.**
- Typically, a service creates its AWS managed CMK in your account when you set up the service or the first time you use the CMK.

AWS KMS – Default Master Key vs. CMKs

- The AWS services that integrate with AWS KMS can use it in many ways.
  - Some services create AWS managed CMKs in your account.
  - Other services require that you specify a customer managed CMK that you have created.
  - Others support both types of CMKs to allow you the ease of an AWS managed CMK or the control of a customer-managed CMK.

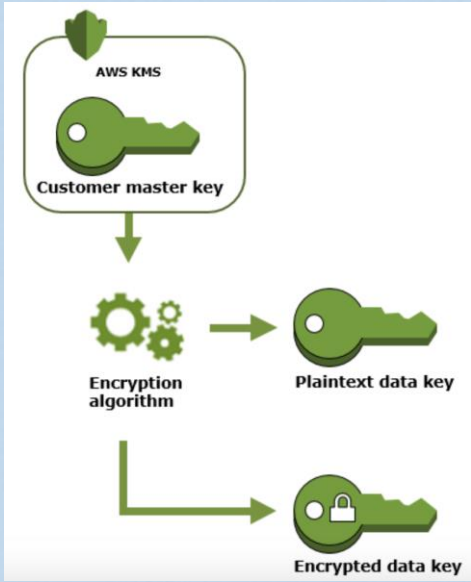
- You have the option of selecting a specific master key to use when you want an AWS service to encrypt data on your behalf.
- A *Default Master* Key (Default CMK) **specific to each service is created in your account** as a convenience the first time you try to create an encrypted resource.
  - This key is managed by AWS KMS but you can always audit its use in AWS CloudTrail.
  - AWS will update the policies on default master keys as needed to enable new features in supported services automatically.
- You can alternately create a **custom master key in AWS KMS** that you can then use in your own applications or from within a supported AWS service.
- AWS does not modify policies on keys you create.

AWS Key Management Service

- Data Keys and CMKs
- Envelope Encryption, Key Rotation

AWS KMS Data (or Object) Keys

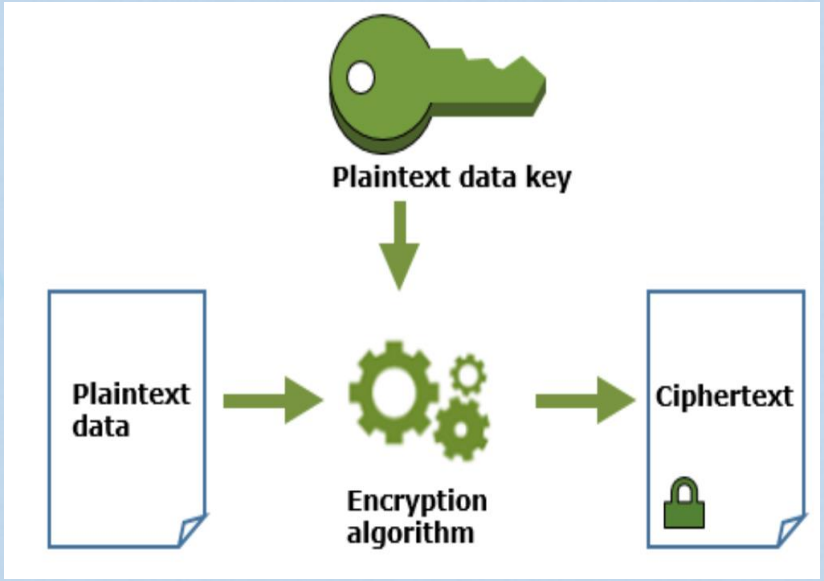
- Data keys are encryption keys that you can use to encrypt data, including large amounts of data and other data encryption keys.
- You can use AWS KMS customer master keys (CMKs) to generate, encrypt, and decrypt data keys.
- AWS KMS **does not** store, manage, or track your data keys, or perform cryptographic operations with data keys.
- You must use and manage data keys outside of AWS KMS.
- To create a data key, call the GenerateDataKey operation.
  - AWS KMS uses the CMK that you specify to generate a data key.
    - The operation returns
      - **A plaintext copy of the data key** and
      - **An Encrypted copy of the data key, encrypted under the specified CMK.**
- AWS KMS also supports the GenerateDataKeyWithoutPlaintext operation, which returns only an encrypted data key.
- When you need to use the data key, you need to request AWS KMS to decrypt it



AWS KMS – Encrypting/Decrypting 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.

- **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.



AWS KMS – Envelope Encryption

- When you encrypt your data, your data is protected, but you must protect your encryption key.
- One strategy is to encrypt the data key, which is used to encrypt/decrypt your data

- **Envelope encryption**
  - Is the practice of encrypting plaintext data with a data key, and then encrypting the data key under another key.
  - You can even encrypt the data encryption key under another encryption key, and encrypt that encryption key under another encryption key.
  - But, eventually, **one key must remain in plaintext** so you can decrypt the keys and your data.

AWS KMS – Key Policies

Key Policies

- When you create a CMK, you determine who can use and manage that CMK.
- These permissions are contained in a document called the key policy
- You can use the key policy to add, remove, or modify permissions at any time for a customer-managed CMK, but you cannot edit the key policy for an AWS-managed CMK.

Auditing CMK Usage

- You can use AWS CloudTrail to audit key usage.
- The CloudTrail log files include all AWS KMS API requests made with the AWS Management Console, AWS SDKs, and command line tools, as well as those made through integrated AWS services.
- You can use these log files to get information about when the CMK was used, the operation that was requested, the identity of the requester, the IP address that the request came from, and so on.

AWS KMS – Rotating Customer Master Keys (CMKs)

Cryptographic best practices discourage extensive reuse of encryption keys.

- To create new cryptographic material for your AWS Key Management Service (AWS KMS) customer master keys (CMKs),
  - You can create new CMKs, and then change your applications or aliases to use them.
  - Or, you can enable automatic key rotation for an existing CMK.
    - When you enable automatic key rotation for a customer managed CMK,
      - AWS KMS generates new cryptographic material for the CMK every year.
- AWS KMS also saves the CMK's older cryptographic material so it can be used to decrypt data that it encrypted.
- Automatic key rotation has no effect on the data that the CMK protects.
  - It does not rotate the data keys that the CMK generated or re-encrypt any data protected by the CMK, and it will not mitigate the effect of a compromised data key.

Manually Rotating CMKs

- You might decide to create a new CMK and use it in place of the original CMK.
- This has the same effect as rotating the key material in an existing CMK, so it's often thought of as manually rotating the CMK.
- Manual rotation is a good choice when you want to control the key rotation schedule.
- It also provides a way to rotate CMKs with imported key material.

Encrypting Data at Rest

Encrypting Data at Rest

AWS EBS and KMS

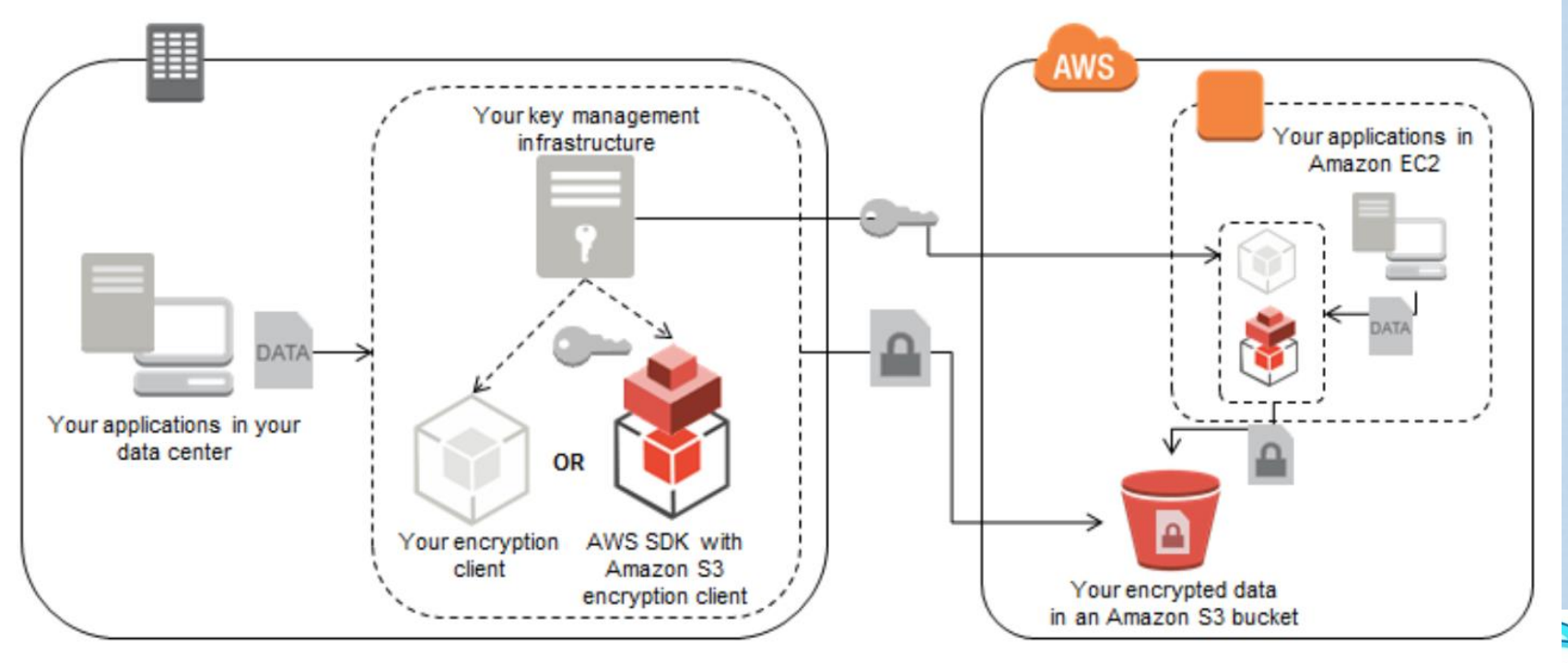
Amazon EBS Encryption

- When encrypting an EBS volume, You can use a custom Customer Master Key (CMK).
- If you do not specify a custom CMK, Amazon EBS uses the **AWS-managed CMK** for Amazon EBS in your account.
- If there is no AWS-managed CMK for Amazon EBS in your account, Amazon EBS creates one.
- 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.
- 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.
- 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.

Encrypting Data at Rest

Encryption Model A – AWS S3

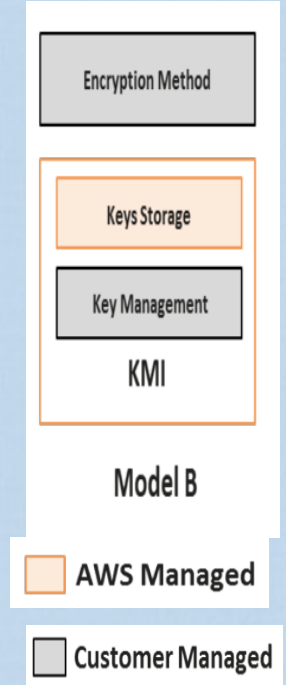
Amazon S3 client-side encryption from on-premises system or from within the client's Amazon EC2 application



Encrypting Data at Rest

Encryption Model B

- An HSM can be used to generate and store key material and can perform encryption and decryption operations,
  - But it does not perform any key lifecycle management functions (e.g., access control policy, key rotation).
    - This means that a compatible KMI might be needed in addition to the AWS CloudHSM appliance before deploying your application.
    - The KMI you provide can be deployed either on-premises or within Amazon EC2 and can communicate to the AWS CloudHSM instance securely over SSL to help protect data and encryption keys.



Encrypting Data at Rest

Encryption Model C

- In this model, AWS provides server-side encryption of your data, transparently managing the encryption method and the keys.

- **Key Management Services (KMS)**
  - AWS KMS is a managed encryption service that lets you provision and use keys to encrypt your data in AWS services and your applications.
  - Master keys in AWS KMS are used in a fashion similar to the way master keys in an HSM are used.
    - After master keys are created, they are designed to never be exported from the service.
    - Data can be sent into the service to be encrypted or decrypted under a specific master key under you account.
    - This design gives you centralized control over who can access your master keys to encrypt and decrypt data, and it gives you the ability to audit this access.

Encrypting Data at Rest

Encryption Model C – S3

- There are three ways of encrypting your data in Amazon S3 using server-side encryption.

1. **Server-side encryption: SSE-S3**
2. **Server-Side Encryption using Customer Keys (SSE-C)**
3. **Server-Side Encryption using AWS KMS (SSE-KMS)**

Data Encryption at Rest

"s3:x-amz-server-side-encryption": "AES256"

Encryption Model C – S3 SSE-S3

1. **Server-side encryption: SSE-S3**
   - KMS generates this **data key** and encrypts it using the **master key that you** specified earlier;
   - KMS then returns this **encrypted data key** along with the **plaintext data key** to Amazon S3.
   - Amazon S3 encrypts the object using the **plaintext data key first ,** and then stores the now encrypted object (along with the encrypted object key) and deletes the plaintext object key from memory.
   - To retrieve this encrypted object, Amazon S3 sends the encrypted data key to AWS KMS.
   - AWS KMS decrypts the data key using the correct master key and returns the decrypted (plaintext) object key to S3.
   - With the plaintext object key, S3 decrypts the encrypted object and returns it to you.
   - Each object is encrypted with a unique data key,
     - This key is encrypted with a periodically rotated key managed by AWS S3.
     - Amazon S3 server-side encryption uses 256-bit Advanced Encryption Standard (AES) keys for both object and master keys.

- This feature is offered at no additional cost beyond what you pay for using Amazon S3.
- If you want SSE-S3 then the **x-amz-server-side-encryption** header must define SEE-S3.

  “s3:x-amz-server-side-encryption":"aws:AES256”

Data Encryption at Rest

Encryption Model C – S3 SSE-C

2. **Server-side encryption using customer provided keys: (SSE-C)**
   - Clients can use their own encryption key while uploading an object to Amazon S3.
   - This encryption key is used by Amazon S3 to encrypt your data using AES-256.
   - S3 does not store the key, after the object is encrypted, the encryption key supplied by the client is deleted from the Amazon S3 system that used it to protect the client's data.
   - When the client retrieves this object from Amazon S3, they must provide the same encryption key in the request.
     - Amazon S3 verifies that the encryption key matches,
     - Decrypts the object, and returns the object to the requester.

- This feature is offered at no additional cost beyond what you pay for using Amazon S3.
- If you want to manage your own encryption keys, provide all the following headers in the request.
  - x-amz-server-side-encryption-customer-algorithm
  - x-amz-server-side-encryption-customer-key
  - x-amz-server-side-encryption-customer-key-MD5

Data Encryption at Rest

Encryption Model C – S3 SSE-KMS

3. Server-side encryption using KMS:

- You can encrypt the data in Amazon S3 by defining an AWS KMS master key within your account that you want to use to encrypt the unique object (data) key that will ultimately encrypt your object (data).
- When you upload your object, a request is sent to KMS to create an object key.
- The first time you add an SSE-KMS–encrypted object to a bucket in a region, a default CMK is created for you automatically.
  - This key is used for SSE-KMS encryption unless you select a CMK that you created separately using AWS Key Management Service.
  - Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, and define access controls, and to audit the encryption keys used to protect your data.
- Amazon S3 supports bucket policies that you can use if you require server-side encryption for all objects that are stored in your bucket.

- For SSE to be requested in an API call, the request has to include the **x-amz-server-side-encryption** header requesting server-side encryption
  - If you want SSE-KMS then the **x-amz-server-side-encryption** header must define SEE-KMS.

  “s3:x-amz-server-side-encryption":"aws:kms”

Data Encryption at Rest

Encryption Model C – AWS Glacier

Amazon Glacier

- Before it's written to disk, data is always automatically encrypted using 256-bit AES keys unique to the Amazon Glacier service that are stored in separate systems under AWS control.
- This feature is offered at no additional cost beyond what you pay for using Amazon Glacier.

AWS Storage Gateway

- The AWS Storage Gateway transfers your data to AWS over SSL and stores data encrypted at rest in Amazon S3 or Amazon Glacier using their respective server side encryption schemes.

Amazon EMR

- EMR supports the ability to request Amazon S3 to use server-side encryption when it writes EMR data to an Amazon S3 bucket you manage.
- This feature is offered at no additional cost beyond what you pay for using Amazon S3 to store your Amazon EMR data.

AWS CLOUDTRAIL

AWS CloudTrail

Introduction and Benefits

CloudTrail Benefits

- You can identify:
  - Who or what took which action,
  - What resources were acted upon,
  - When the event occurred, and other details to help you analyze and respond to activity in your AWS account.

This can benefit in the following areas

- Security
  - Visibility into your AWS account activity is a key aspect of security best practices.
- Tracking changes in an AWS environment
  - You can use CloudTrail to view, search, download, archive, analyze, and respond to account activity across your AWS infrastructure.
- Compliance and auditing
- Operational and troubleshooting support

CloudTrail Event History and Trails

- Event history allows you to view, search, and download the past 90 days of activity in your account.
- You can create a **CloudTrail trail** to archive, analyze, and respond to changes in your AWS resources.
- CloudTrail logging, which is basically, sending the CloudTrail events to a bucket **is not enabled by default.**
  - You need to create a Trail and define a bucket, then CloudTrail will send events to this bucket, i.e will start logging the identified/selected events.
- A trail is a configuration that enables delivery of events to an Amazon S3 bucket that you specify.
- You can deliver and analyze events in a trail with Amazon CloudWatch (CW) Logs and CW Events.
- You can create a trail with the CloudTrail console, the AWS CLI, or the CloudTrail API.

Using AWS CloudTrail with Interface VPC Endpoints

- If you use Amazon VPC to host your AWS resources, you can establish a private connection between your VPC and AWS CloudTrail.

CloudTrail Event History and Trails – Trail Types

- **A trail that applies to all regions (Recommended by AWS)**
  - When you create a trail that applies to all regions,
    - CloudTrail records events in each region and delivers the CloudTrail event log files to an S3 bucket that you specify.
      - This is effectively like creating the trail in each of these regions
  - If a region is added after you create a trail that applies to all regions,
    - That new region is automatically included, and events in that region are logged.
  - This is the **default option** when you create a trail in the **CloudTrail console.**
- **A trail that applies to one region**
  - When you create a trail that applies to one region,
    - CloudTrail records the events in that region only.
    - It then delivers the CloudTrail event log files to an Amazon S3 bucket that you specify.

Note

- For both types of trails, you can specify an Amazon S3 bucket from any region.

Advantages of All Regions Trails

- A trail that applies to all regions has the following advantages:
  - The configuration settings for the trail apply consistently across all regions.
  - Receiving CloudTrail events from all regions in a single S3 bucket and, optionally, in a CloudWatch Logs log group.
  - Managing trail configuration for all regions from one location.
  - Immediately receiving CloudTrail events from a new region when launched.
  - Ability to create trails in regions that you don't use often to monitor for unusual activity.
  - If CloudTrail is configured to use an Amazon SNS topic for the trail, SNS notifications about log file deliveries in all regions are sent to that single SNS topic.

Multiple Trails per Region

- If you have different but related user groups, such as developers, security personnel, and IT auditors, you can create multiple trails per region.
  - This allows each group to receive its own copy of the log files.
- CloudTrail supports five trails per region.
  - A trail that applies to all regions counts as one trail in every region.

Encryption of CloudTrail Events' Logging

- By default, the log files delivered by CloudTrail to your bucket are encrypted by Amazon server-side

  encryption with Amazon S3-managed encryption keys (SSE-S3).

- To provide a security layer that is directly manageable, you can instead use server-side encryption with AWS KMS–managed keys (SSE-KMS) for your CloudTrail log files.

Note

- Enabling server-side encryption encrypts the log files but not the digest files with SSE-KMS. Digest files are encrypted with Amazon S3-managed encryption keys (SSE-S3).

Monitoring and Notifications

SNS notifications:

- If you want notifications about log file delivery and validation,
  - Create and subscribe to an Amazon SNS topic to receive notifications about log file delivery to your bucket.
    - Amazon SNS can notify you in multiple ways, including programmatically with Amazon Simple Queue Service.

Monitor events with CloudWatch Logs

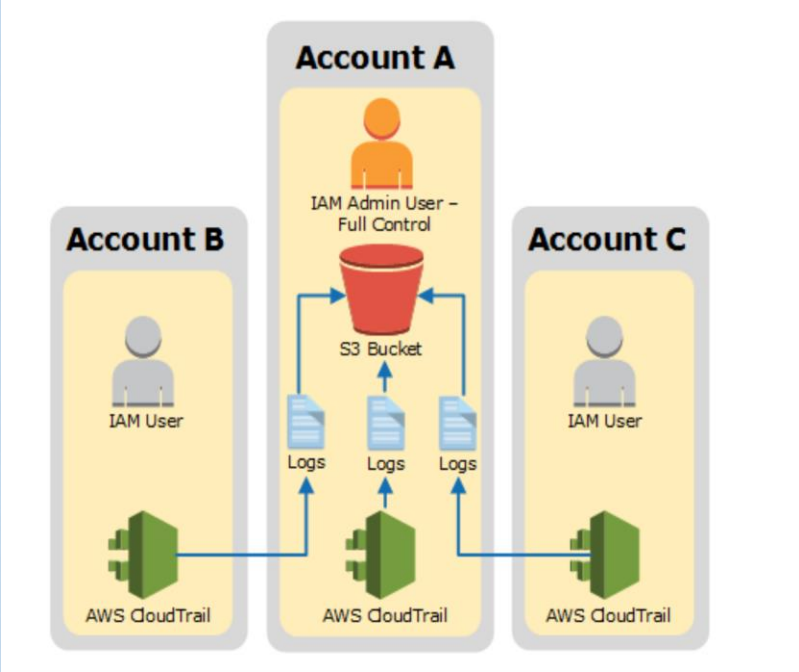
- You can configure your trail to send events to CloudWatch Logs. You can then use CloudWatch Logs to monitor your account for specific API calls and events

AWS CloudTrail

Central Logging in Multi-Accounts & CloudTrail

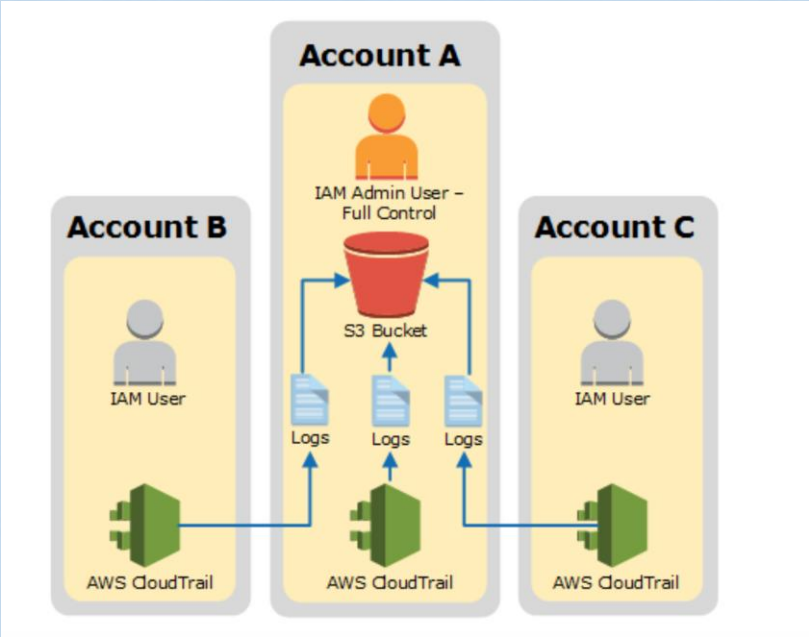
Receiving CloudTrail Log files from multiple AWS Accounts

- CloudTrail can deliver log files from multiple AWS accounts into a single Amazon S3 bucket.
- For example, you have three AWS accounts with account IDs A, B, C
  - You want to configure CloudTrail to deliver log files from all of these accounts to a bucket belonging to account A.
  - To accomplish this:
    - Turn on CloudTrail in the account where the destination bucket will belong, Account A
    - Do not turn on CloudTrail in any other accounts yet.



Receiving CloudTrail Log files from multiple AWS Accounts

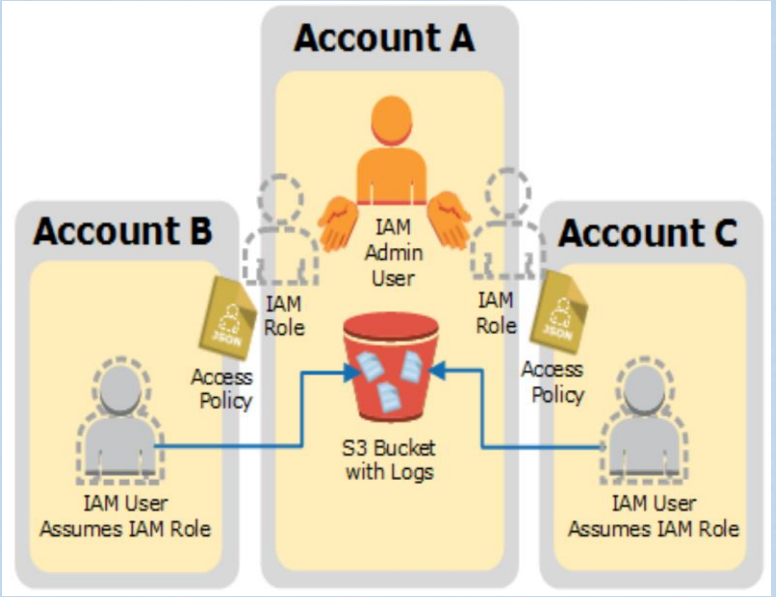
- Update the bucket policy on your destination bucket to grant cross-account permissions to CloudTrail
  - For a bucket to receive log files from multiple accounts, its bucket policy must grant CloudTrail permission to write log files from all the accounts you specify.
  - This means that you must modify the bucket policy on your destination bucket to grant CloudTrail permission to write log files from each specified account.
- Turn on CloudTrail in the other accounts B and C
- Configure CloudTrail in these accounts to use the same bucket belonging to account A



Sharing CloudTrail Log files between Accounts

Let's say in the previous example, now accounts B and C need access to their logs they uploaded in Account A's bucket.

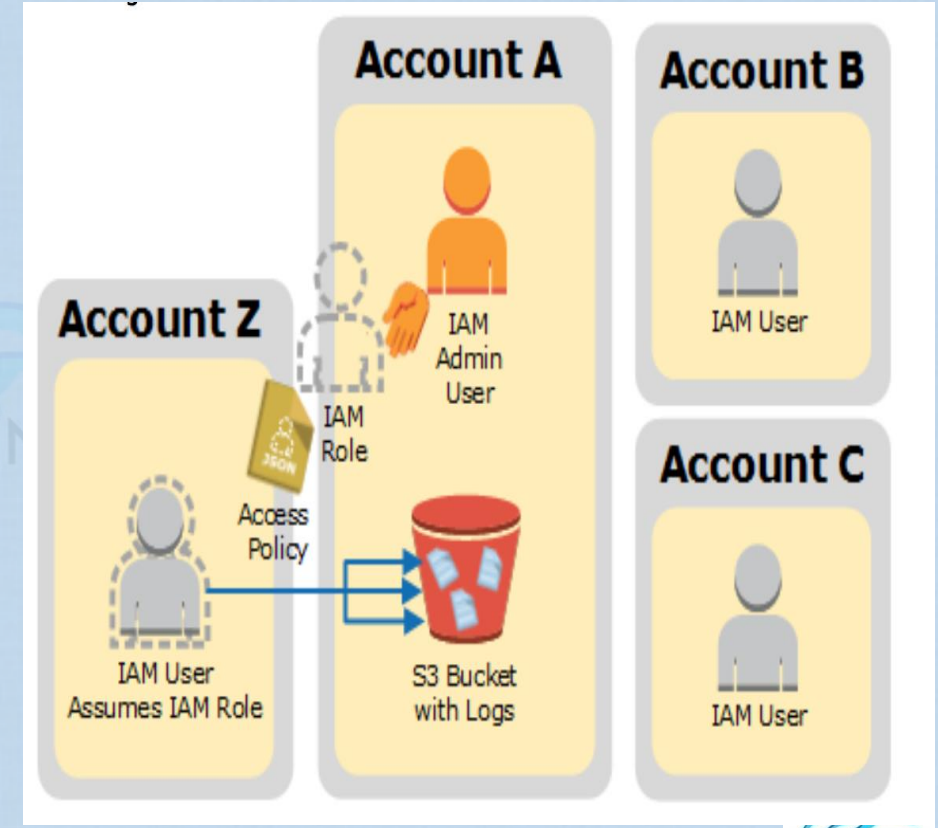
- To share log files between multiple AWS accounts, you must perform the following general steps.
  - In Account A, Create an IAM role for each account that you want to share log files with (B, C).
  - For each of these IAM roles, create an access policy that grants **read-only** access to the respective account (B or C)
  - An IAM user in each account (B and C) programmatically assume the appropriate role and retrieve the log files.



Sharing CloudTrail Log files between Accounts

If you want to grant/share logs with a third party (Auditor, Log analysis 3rd party provider, ..etc)

- You deal with it exactly the same
  - Create a Trust policy to the third party AWS account, in which you allow that account to assume an IAM role
    - The IAM role has an attached permission policy to define what is/isn't allowed to access in the CloudTrail log bucket
    - The third party account needs to Assume the IAM role and then will have access to the logs
    - It has to be confined to READ only access (GET and LIST operations)



AWS CloudTrail

Log File Integrity Validation

CloudTrail Log File Integrity Validation – What is it

- Is the ability of Amazon CloudTrail to determine whether a log file was modified, deleted, or unchanged after CloudTrail trail has delivered it to your S3 bucket.
- This feature is built using industry standard algorithms: SHA-256 for hashing and SHA-256 with RSA for digital signing.
  - This makes it computationally infeasible to modify, delete or forge CloudTrail log files without detection.
- You can use the AWS CLI to validate the files in the location where CloudTrail delivered them.
- Benefits of this feature:
  - Validated log files are invaluable in security and forensic investigations.
  - A validated log file enables you to assert positively that the log file itself has not changed, or
  - Whether a particular user credentials performed specific API activity.
  - It lets you know if a log file has been deleted or changed, or
  - Assert positively that no log files were delivered to your account during a given period of time.

CloudTrail Log File Integrity Validation – How It Works

- When the validation feature is enabled
  - CloudTrail creates a hash for every log file that it delivers.
  - Every hour, CloudTrail creates and delivers a file that references the log files for the last hour and contains a hash of each.
    - This file is called a digest file.
    - CloudTrail signs each digest file using the private key of a public and private key pair.
    - After delivery, the public key can be used to validate the digest file.
    - CloudTrail uses different key pairs for each AWS region.
- The digest files are delivered to the same Amazon S3 bucket associated with the trail as your CloudTrail log files.
- Each digest file also contains the digital signature of the previous digest file if one exists.
- The signature for the current digest file is in the metadata properties of the digest file Amazon S3 object

CloudTrail Log File Integrity Validation – How It Works

- If the log files are delivered from all regions or from multiple accounts into a single Amazon S3 bucket,
  - CloudTrail will deliver the digest files from those regions and accounts into the same bucket.
  - The digest files are put into a folder separate from the log files.
    - This separation of digest files and log files enables you to enforce granular security policies and permits existing log processing solutions to continue to operate without modification.

- The CloudTrail log files and digest files can be stored securely in Amazon S3 or Glacier securely, durably and inexpensively for an indefinite period of time.

- To enhance the security of the digest files stored in Amazon S3, you can use Amazon S3 MFA Delete protection

- AWS Management Console, APIs or CLI can be used to enable the feature

CloudTrail Log File Integrity Validation – Validating Log files integrity

- To validate the integrity of CloudTrail log files the AWS CLI or create 3rd party solution can be used.
- The AWS CLI will validate files in the location where CloudTrail delivered them.
- Validation for logs that have been moved to a different location, either in Amazon S3 or elsewhere,
  - Customers need to create their own validation tools.

AWS DDOS

AWS DDoS Protection

Mitigation Techniques

DDoS – Mitigation Techniques

- You do not pay for inbound data transfer in AWS, so you are not charged for inbound DDoS attack traffic
- AWS offers some DDoS mitigation techniques that included automatically with AWS services.
- Clients can improve their DDoS resilience further by using an AWS architecture with specific services and by implementing additional best practices (Route53, CloudFront, WAF)
- **AWS Shield Standard**,
  - Is an automated AWS protection service at no additional charge.
  - AWS Shield Standard defends against most common, frequently occurring network and transport layer DDoS attacks that target your web site or applications.
  - It is offered on all AWS services and in every AWS Region.
  - It provides protection against many common infrastructure layer attacks

DDoS – Mitigation Techniques

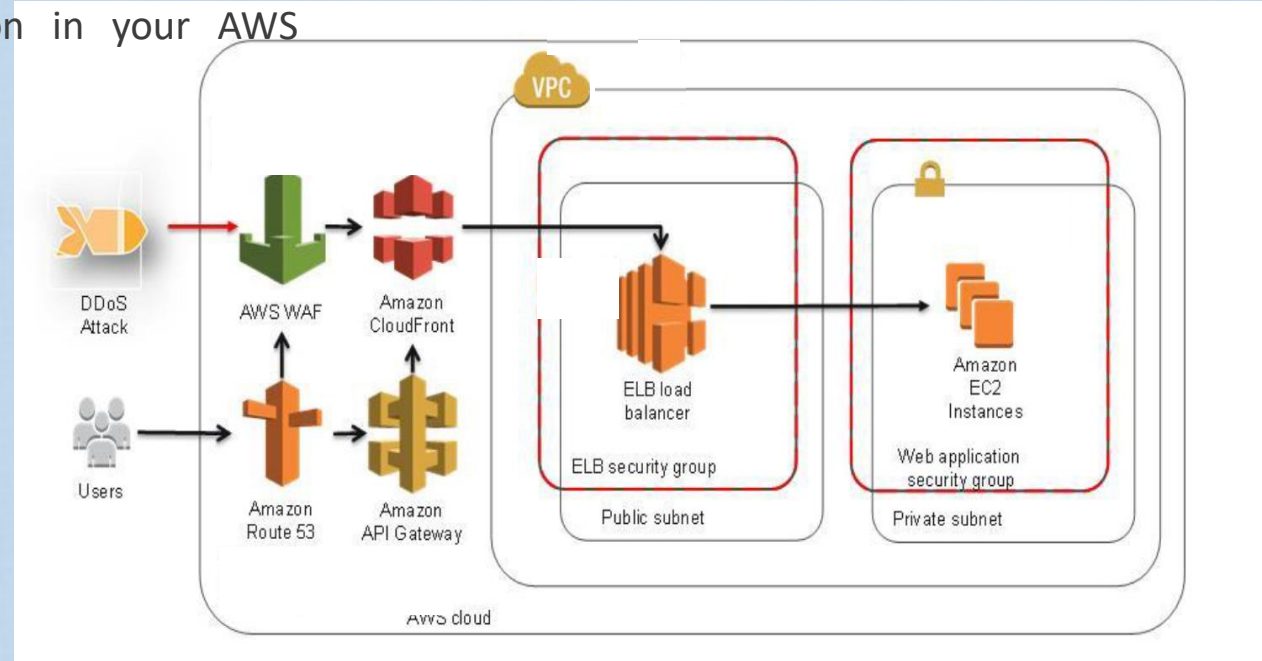
- Build resilience – adaptive, scalable, layered

- Proactively employ multiple techniques
  - Minimize attack surface area
  - Learn normal behavior
  - Be ready to scale
  - Safeguard expensive resources that are hard to scale
  - Have a plan to deal with DDoS

AWS DDoS

DDoS-resilient reference architecture w/ AWS Global Edge Network services

- Combining these different AWS edge-based services allows you to have a far better DDoS mitigation/protection in your AWS infrastructure



DDoS – Mitigation Techniques – Route53 and CloudFront

- **To further protect from all known Infrastructure layer attacks,**
  - AWS recommends to leverage AWS services that operate from edge locations, like Amazon CloudFront and Amazon Route 53.
  - Using these services – part of the AWS Global Edge Network – can improve the DDoS resilience of your application when you serve web application traffic from edge locations distributed around the world.

- **Benefits of using Amazon CloudFront and Amazon Route 53 include:**
  - AWS Shield DDoS mitigation systems that are integrated with AWS edge services, reducing time-to-mitigate from minutes to sub-second.
  - Stateless SYN Flood mitigation techniques that proxy and verify incoming connections before passing them to the protected service.
  - Automatic traffic engineering systems that can disperse or isolate the impact of large volumetric DDoS attacks.
  - Application layer defense when combined with AWS WAF

DDoS Mitigation – AWS Shield Advanced Service

- To improve your AWS hosted, or elsewhere, infrastructure and applications' readiness to respond to and mitigate DDoS attacks, you can subscribe to AWS Shield Advanced.

- It is an optional DDoS mitigation service that can help you to protect an application hosted on any AWS Region or hosted outside of AWS.

- The service is available globally for:
  - Amazon CloudFront and
  - Amazon Route 53.

- It's also available in select AWS Regions for:
  - ELB (CLB, ALB)
  - Using Advanced shield with Elastic IP Addresses (EIPs), allows you to protect Network Load Balancer (NLBs) or Amazon EC2 instances.

AWS DDoS Protection

Infrastructure and Application Layer Defense

Infrastructure Layer Defense

- EC2 – Scale Vertically (Instance families) and Horizontally (Auto Scaling)
  - Enhanced networking and high-speed network throughput (up to 25Gbps)

- ELB and scaling to not overload backend instances
  - You can reduce the risk of overloading your application by distributing traffic across many backend instances
    - ALB can detect application layers attacks and scale to absorb its traffic
    - ELB does not support UDP and will drop any UDP traffic, protecting from UDP attacks

- Using Route53 can protect you from DNS attacks to your own or 3rd party DNS system

- CloudFront will help you mitigate attacks through minimizing requests to your origin server, and with its security features.

Application Layer Defense

- Using Amazon Cloudfront and Amazon WAF you can protect against application layer attacks

- Detect and Filter Malicious requests
  - Cloudfront:
    - Amazon CloudFront allows you to cache static content and serve it from AWS edge locations,
      - This can help reduce load on your origin server
      - Cloudfront can close slow reading or writing attackers
  - Web Application Firewall
    - You can configure Web ACLs on your Cloudfront distribution or ALB, which has advanced application layer attacks filtering capabilities based on malicious requests.

- Scale to absorb,
  - Use ELB and Autoscaling to scale your fleet in response to attacks (CloudWatch Alarms)

AWS DDoS Protection

Attack Surface Reduction

Attack Surface Reduction

- This means limiting your application's internet exposure,
- It is well known that resources that are not exposed to the public internet could be more difficult to attack, hence, limiting the internet exposure
- Strategy to minimize the Attack surface area
  - Remove or Obfuscate necessary Internet entry points to the level that untrusted end users cannot access them (make them hard to find)
  - Reduce the number of necessary Internet entry points,
    - IP addresses and ports on your infrastructure that can be accessed from the internet
  - Eliminate non-critical Internet entry points
  - Don't expose back end servers,
  - Separate end user traffic from management traffic,
  - Decouple Internet entry points to minimize the effects of attacks, spread your applications on different servers

Attack Surface Reduction

Obfuscating AWS Resources

- You can use ELB to be the only communication channel
- Hide your Instances from the Internet - Use Bastion hosts to access your EC2 instances from the internet
- Use NAT Gateways to provide internet access from within your VPC, but not the other way
- Use Sec Groups and NACLs to prevent unauthorized access to your EC2 instances
- Protect your Origin servers and limit access to them using Cloudfront
- Protect API endpoints, -> Using API Gateway in front of your applications

AWS DDoS

Scalable, Secure, Well-Monitored, DDoS protected Applications on AWS

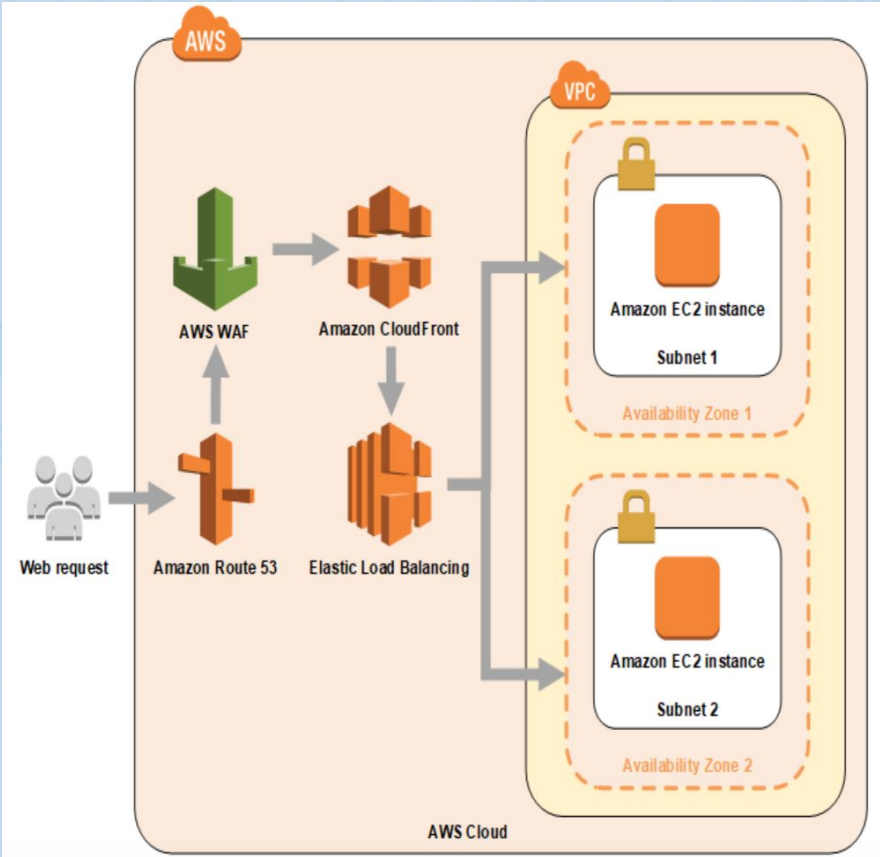
Steps to build such an Application on AWS

- Use globally distributed services like CloudFront and Route53
  - Built-in protection against L3 and 4 DDoS attacks
- Use Elastic Load Balancer
  - Does not accept UDP and accepts only well formed packets
- Build DDoS protected application
  - Scale compute to absorb attack traffic -> Instance scaling and auto scaling
- Hide instances from the Internet
  - Use Sec Groups and NACLs in a VPC
- Well-monitor your application
  - Enable CloudWatch monitoring based on multiple metrics, create dashboards, interate with SNS
  - Enable logging for deeper analysis (CloudTrail, Kinesis, and Redshift can help in real time analysis)
- Use WAF to baseline traffic and protect against attacks

AWS WEB APPLICATION FIREWALL (WAF)

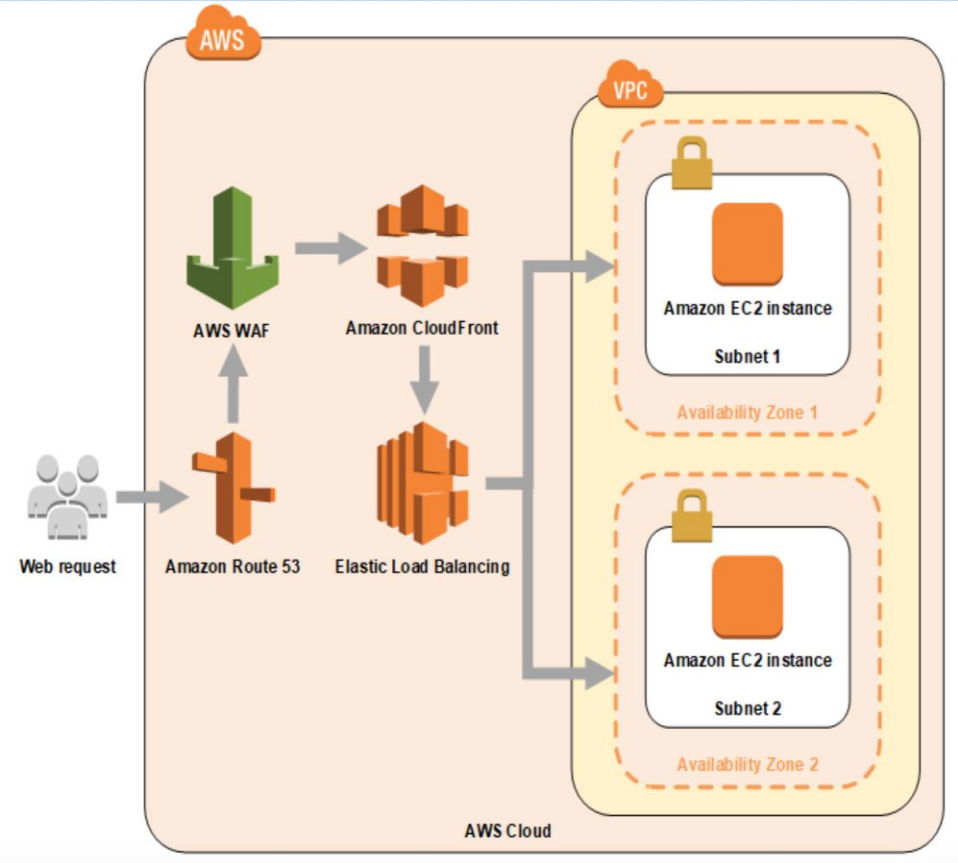
AWS WAF

- AWS WAF is a web application firewall that lets you monitor the **HTTP** and **HTTPS** requests that are forwarded to Amazon CloudFront or an Application Load Balancer.
- AWS WAF also lets you control access to your content, Based on conditions that you specify.
- Using AWS WAF you control whether CloudFront or an Application Load Balancer responds to requests either with the requested content or with an HTTP 403 status code (Forbidden).
  - You also can configure CloudFront to return a custom error page when a request is blocked.



AWS WAF

- When used with CloudFront, it can also protect websites hosted off AWS (Custom Origins of CloudFront)
- Supports IPv6,
- Integrated with CloudTrail for logging AWS WAF API Calls



Example of App attacks Blocked by AWS WAF

- **Cross Site Scripting** – XSS
  - It is a type of injection attacks,
  - Attackers try to inject malicious scripts into trusted websites
  - *XSS attacks* occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.
    - This malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site.

- **SQL injection Attacks** –
  - A SQL injection attack consists of injection of a SQL query via the input data from the client to the application.
    - If successful, a successful SQL injection exploit can:
      - Read sensitive data from the database,
      - Modify database data (Insert/Update/Delete),
      - Execute administration operations on the database

AWS WAF – Web Access Control Lists (WACLs)

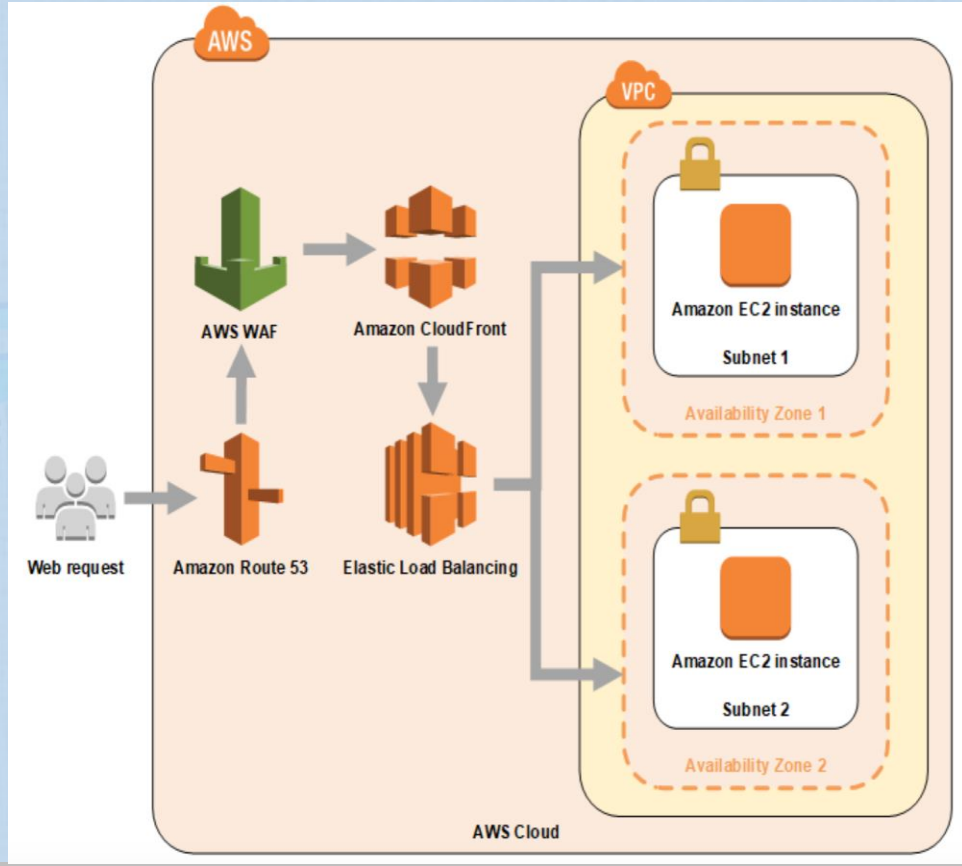
- Web ACLs consists of a collection of configured rules
- **For each rule, the WACL decision can be - allow, block or count based on conditions you define.**

- **How it applies the rules:**
  - WAF compares a request against the included rules in the order listed,
  - When a web request matches all of the conditions in a rule, WAF immediately takes the action—allow or block—and it aborts (will not check the remaining rules).
  - If the action was Count, then it will increase the counter and continue to inspect the request against remaining rules.

- **Default action**
  - If the request did not match any rule, this would instruct the WAF to either allow or block it.
    - Count can not be specified as the default action

AWS WAF – How it works

- As the underlying service receives requests for your web sites, it forwards those requests to AWS WAF for inspection against your rules.
- Once a request meets a condition defined in your rules,
  - AWS WAF instructs the underlying service to either block or allow the request based on the action you define.
  - You can whitelist and blacklist IP addresses



AWS WAF

- WAF & CloudFront
  - When you use AWS WAF on Amazon CloudFront, your rules run in all AWS Edge Locations, located **Globally around the world** close to your end users.
    - This means security doesn't come at the expense of performance.
    - Blocked requests are stopped before they reach your web servers.

- When you use AWS WAF on Application Load Balancer, your rules **run in region** and
  - Can be used to protect internet-facing as well as internal load balancers.

INTRUSION DETECTION/PREVENTION SYSTEM ON AWS

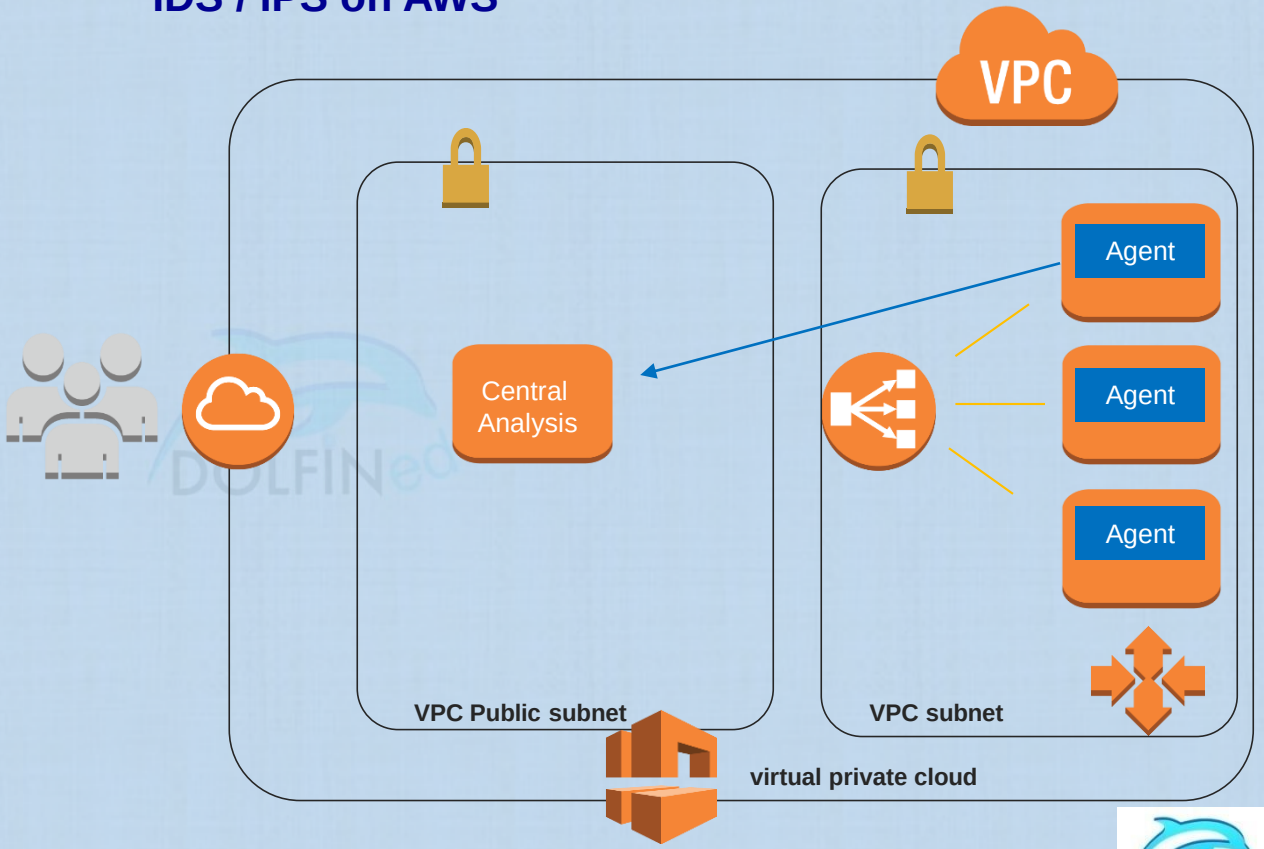
IDS/IPS on AWS

Deployment Scenarios

IDS / IPS on AWS

On host (EC2 instance)

- Monitoring with host based IDS, a HIDS on each EC2 instance
- **Cons:**
  - CPU Intensive
  - Has to replicate to the analysis system which increases the network traffic and associated costs (if outside of AWS)



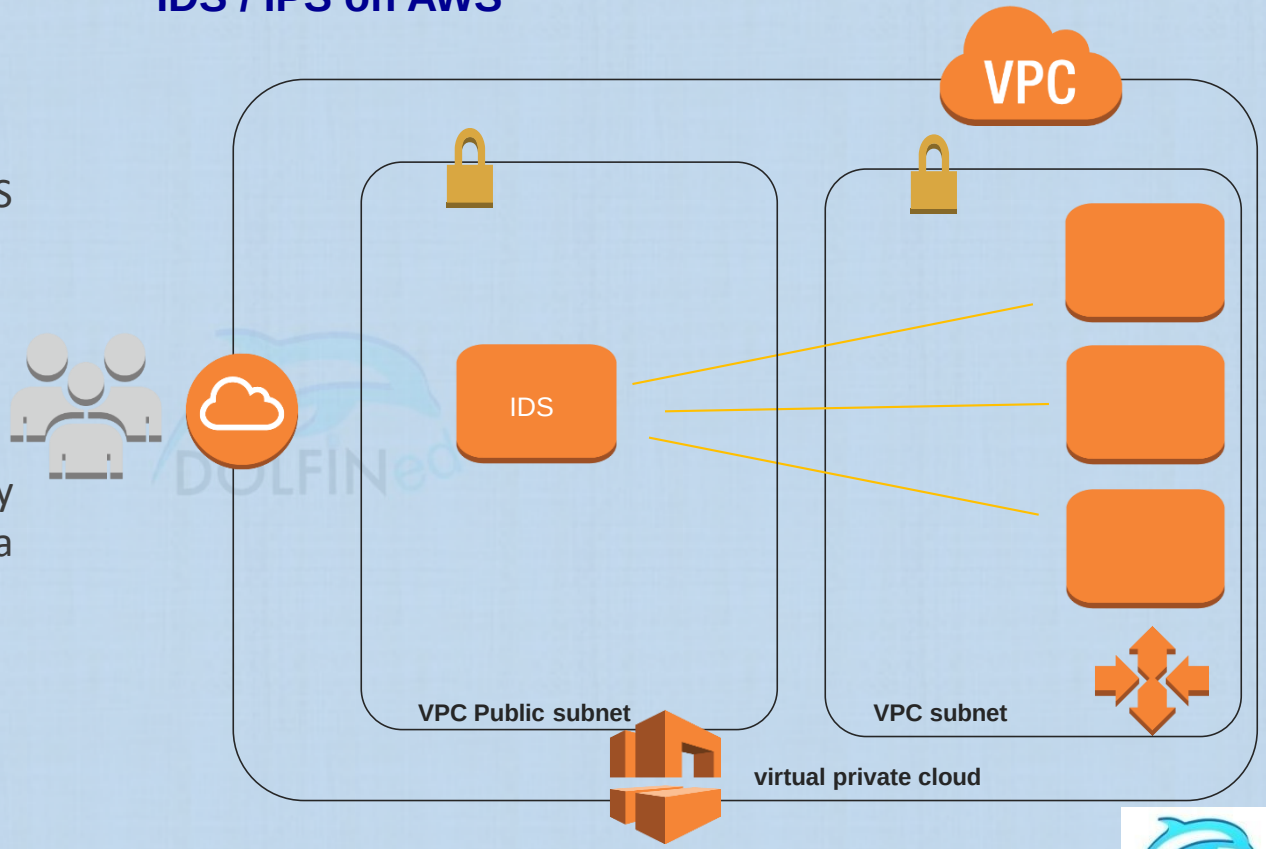
IDS/IPS on AWS

IDS / IPS on AWS

On NAT instance

- Monitoring with an IDS on a NAT/Proxy instance

- **Cons:**
  - Would not scale
  - NAT instance/Proxy could become a bottle neck

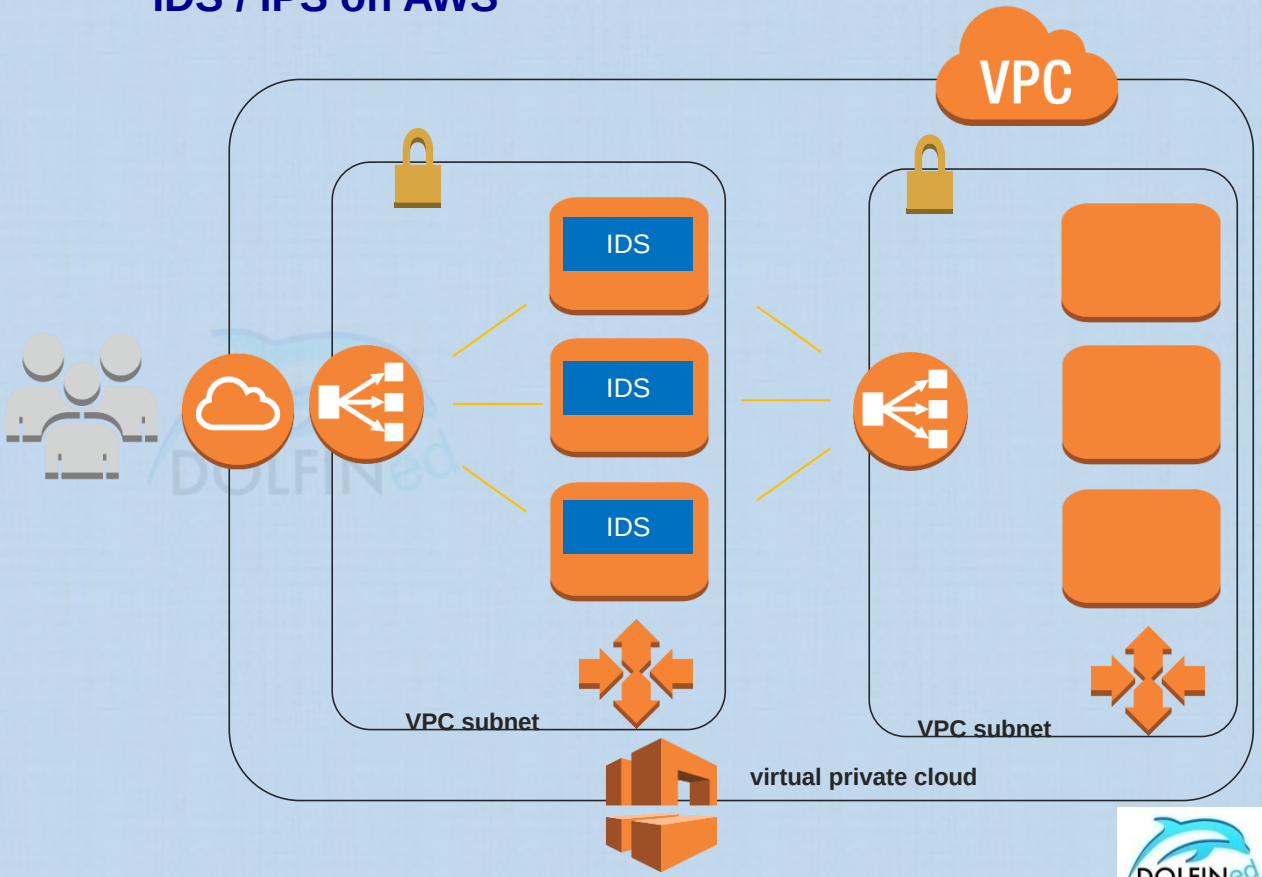


IDS/IPS on AWS

IDS / IPS on AWS

Infrastructure Tier on AWS (DMZ Proxy implementation)

- Similar to AWS WAF Sandwich deployment
- IDS processing and traffic load is now on a separate, scalable, tier in your architecture.
- **Cons:**
  - Cost and administration overhead for this mission critical security tier

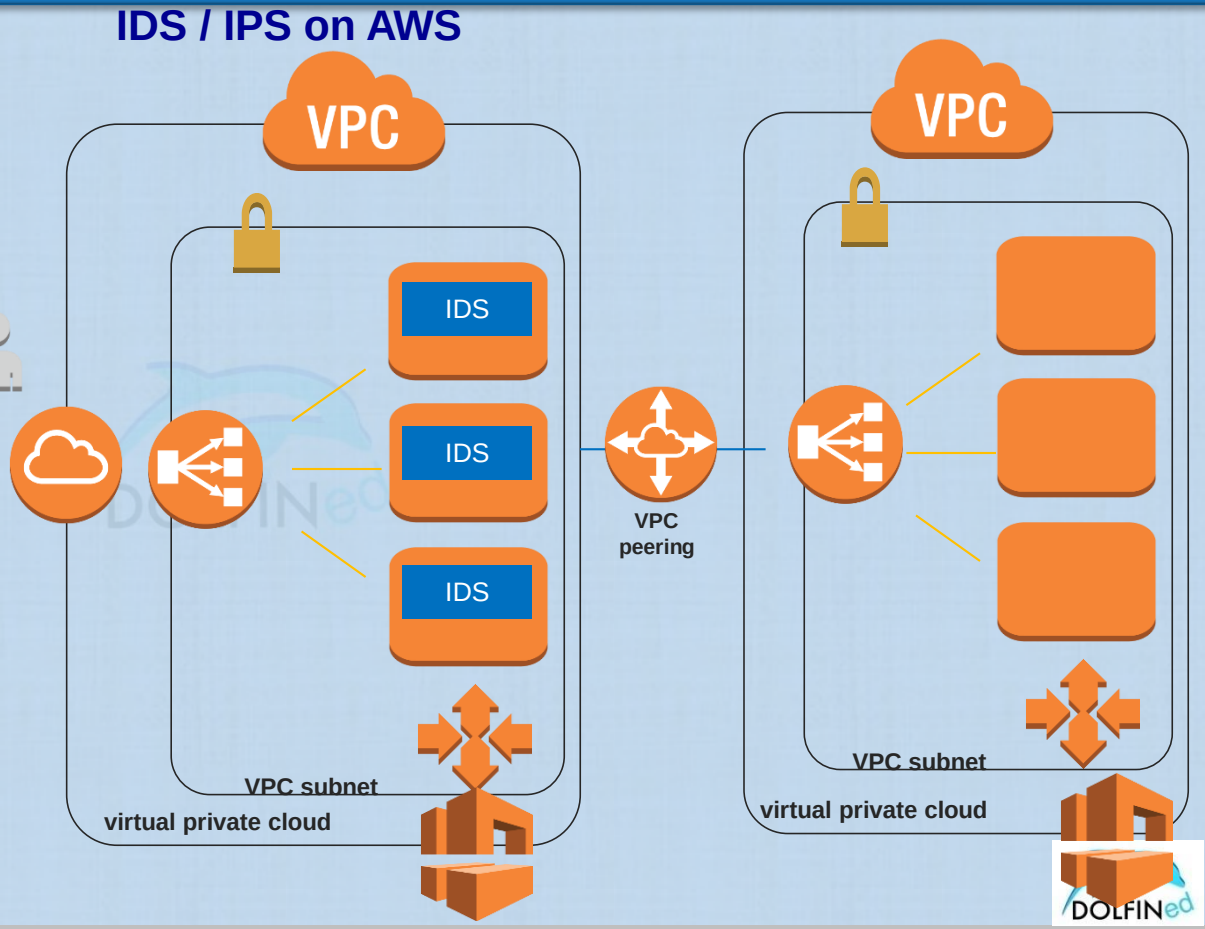


IDS/IPS on AWS

Implement a Security VPC and peer with your Web/App VPC you are protecting

- Scalable IDS/IPS implementation

- **Cons:**
  - Cost and administration overhead for this mission critical security VPC



AMAZON RESOURCE ACCESS MANAGER

AWS Resource Access Manager (RAM)

- What is it?
- How it works
- Benefits
- Resources
- AZ ID
- Monitoring and Logging

AWS Resource Access Manager (RAM) – What is it?

- AWS Resource Access Manager (AWS RAM) enables the sharing of resources with any AWS account or through AWS Organizations.
- For multi-account organizations, resources can be created centrally, and AWS RAM can be used to share those resources with/among the accounts.
- There are no additional charges for creating resource shares and sharing your resources across accounts. Resource usage charges vary depending on the resource type.
- AWS RAM can be accessed from RAM Console, Query APIs, AWS CLI, and AWS Tools for Windows PowerShell
- AWS RAM Integrates with AWS CloudWatch Events
- Calls made to the AWS RAM API can be logged using AWS CloudTrail

AWS Resource Access Manager (RAM) – How it works?

- When a resource is shared with another account, then that account is granted access to the resource.
  - Any policies and permissions in that account apply to the shared resource.

- Sharing Resources
  - An account can share resources it owns by creating a resource share. When resource share is created, the principals with whom to share are specified.
    - Principals can be AWS accounts, organizational units, or an entire organization from AWS Organizations.
  - The sharing account retains full ownership of the shared resources.

- Using Shared Resources
  - When the owner of a resource in account A shares it with an account B, Account B can access the resource.
  - Shared resources can be accessed using console, CLI, APIs.
  - The actions that account B users are allowed to perform vary depending on the resource type.
    - All IAM policies and service control policies configured in account B apply,
      - which enables account B to continue to leverage its existing investments in security and governance controls.

AWS Resource Access Manager (RAM) – Benefits

- Reduces operational overhead
  - Create resources centrally and use AWS RAM to share those resources with other accounts. This eliminates the need to provision duplicate resources in every account, which reduces operational overhead.

- Provides security and consistency
  - Govern consumption of shared resources using existing policies and permissions, to achieve security and control. AWS RAM offers a consistent experience for sharing different types of AWS resources.

- Provides visibility and auditability
  - View usage details for shared resources through integration with Amazon CloudWatch and AWS CloudTrail. AWS RAM provides comprehensive visibility into shared resources and accounts.

AWS Resource Access Manager (RAM) – Shareable Resources

- **AWS VPC Resources**
  - VPC sharing allows multiple AWS accounts to create their application resources, such as Amazon EC2 instances, Amazon RDS databases, Amazon Redshift clusters, and AWS Lambda functions, into shared, centrally-managed Amazon VPCs.
  - The account that owns the VPC (owner) shares one or more subnets with other accounts (participants) that belong to the **same organization from AWS Organizations.**
  - The participants can then view, create, modify, and delete their application resources in the subnets shared with them.
    - Participants cannot view, modify, or delete resources that belong to other participants or the VPC owner.
- **Transit Gateway**
  - AWS Resource Access Manager (RAM) can be used to share a transit gateway across accounts or across your organization in AWS Organizations.
- **Route53 Forwarding Rules**
  - Sharing the forwarding rules created using one AWS account with other AWS accounts.
- **License Manager**
  - Sharing license configurations from the AWS Organizations master account to Instance in member accounts

AWS RAM – AZ ID

- To ensure that resources are distributed across the Availability Zones for a Region, AWS independently maps Availability Zones to names for each account.
  - For example, the Availability Zone us-east-1a for an AWS account might not have the same location as us-east-1a for another AWS account.

- To identify the location of an account resources relative to your accounts, the AZ ID must be used instead of the AZ names.
  - AZ ID is a unique and consistent identifier for an Availability Zone.
    - For example, use1-az1 is an AZ ID for the useast-1 Region and it is the same location in every AWS account.