**AWS SECURITY, IDENTITY & COMPLIANCE**

# **AWS Organizations**

AWS Organizations helps you centrally govern your environment as you grow and scale your workloads on AWS.

Organizations helps you to centrally manage billing; control access, compliance, and security; and share resources across your AWS accounts.

Using AWS Organizations, you can automate account creation, create groups of accounts to reflect your business needs, and apply policies for these groups for governance.

You can also simplify billing by setting up a single payment method for all of your AWS accounts.

Through integrations with other AWS services, you can use Organizations to define central configurations and resource sharing across accounts in your organization.

AWS Organizations is available to all AWS customers at no additional charge.

The [**AWS Organizations API**](https://docs.aws.amazon.com/organizations/latest/APIReference/Welcome.html) enables automation for account creation and management.

Available in two feature sets:

* Consolidated billing.
* All features.

By default, organizations support consolidated billing features.

Consolidated billing separates paying accounts and linked accounts.

You can use AWS Organizations to set up a single payment method for all the AWS accounts in your organization through consolidated billing.

With consolidated billing, you can see a combined view of charges incurred by all your accounts.

Can also take advantage of pricing benefits from aggregated usage, such as volume discounts for Amazon EC2 and Amazon S3.

Limit of 20 linked accounts for consolidated billing (default).

Policies can be assigned at different points in the hierarchy.

Can help with cost control through volume discounts.

Unused reserved EC2 instances are applied across the group.

Paying accounts should be used for billing purposes only.

Billing alerts can be setup at the paying account which shows billing for all linked accounts.

#### **Core concepts**

Some of the core concepts you need to understand are listed here:

**AWS Organization –**An organization is a collection of AWS accounts that you can organize into a hierarchy and manage centrally.

**AWS Account –**An AWS account is a container for your AWS resources.

**Master Account –**A master account is the AWS account you use to create your organization.

**Member Account –**A member account is an AWS account, other than the master account, that is part of an organization.

**Administrative Root –**An administrative root is the starting point for organizing your AWS accounts. The administrative root is the top-most container in your organization’s hierarchy.

**Organizational Unit (OU) –**An organizational unit (OU) is a group of AWS accounts within an organization. An OU can also contain other OUs enabling you to create a hierarchy.

**Policy –**A policy is a “document” with one or more statements that define the controls that you want to apply to a group of AWS accounts. AWS Organizations supports a specific type of policy called a Service Control Policy (SCP). An SCP defines the AWS service actions, such as Amazon EC2 RunInstances, that are available for use in different accounts within an organization.

#### **Migrating accounts between organizations**

Accounts can be migrated between organizations.

You must have root or IAM access to both the member and master accounts.

Use the AWS Organizations console for just a few accounts.

Use the AWS Organizations API or AWS Command Line Interface (AWS CLI) if there are many accounts to migrate.

Billing history and billing reports for all accounts stay with the master account in an organization.

Before migration download any billing or report history for any member accounts that you want to keep.

When a member account leaves an organization, all charges incurred by the account are charged directly to the standalone account.

Even if the account move only takes a minute to process, it is likely that some charges will be incurred by the member account.

## Resource Groups

You can use resource groups to organize your AWS resources.

In AWS, a resource is an entity that you can work with.

Resource groups make it easier to manage and automate tasks on large numbers of resources at one time.

Resource groups allow you to group resources and then tag them.

The Tag Editor assists with finding resources and adding tags.

**You can access Resource Groups through any of the following entry points:**

* On the navigation bar of the AWS Management Console.
* In the AWS Systems Manager console, from the left navigation pane entry for Resource Groups.
* By using the Resource Groups API, in AWS CLI commands or AWS SDK programming languages.

A resource group is a collection of AWS resources that are all in the same AWS region, and that match criteria provided in a query.

In Resource Groups, there are two types of queries on which you can build a group.

Both query types include resources that are specified in the format AWS::service::resource.

* **Tag-based –**Tag-based queries include lists of resources and tags. Tags are keys that help identify and sort your resources within your organization. Optionally, tags include values for keys.
* **AWS CloudFormation stack-based –**In an AWS CloudFormation stack-based query, you choose an AWS CloudFormation stack in your account in the current region, and then choose resource types within the stack that you want to be in the group. You can base your query on only one AWS CloudFormation stack.

Resource groups can be nested; a resource group can contain existing resource groups in the same region.

# **AWS KMS**

AWS Key Management Store (KMS) is a managed service that enables you to easily encrypt your data.

AWS KMS provides a highly available key storage, management, and auditing solution for you to encrypt data within your own applications and control the encryption of stored data across AWS services.

AWS KMS allows you to centrally manage and securely store your keys. These are known as customer master keys or CMKs.

You can generate CMKs in KMS, in an AWS CloudHSM cluster, or import them from your own key management infrastructure.

These master keys are protected by hardware security modules (HSMs) and are only ever used within those modules.

You can submit data directly to KMS to be encrypted or decrypted using these master keys.

You set usage policies on these keys that determine which users can use them to encrypt and decrypt data and under which conditions.

KMS is tightly integrated into many AWS services like Lambda, S3, EBS, EFS, DynamoDB, SQS etc.

AWS KMS is integrated with AWS services and client-side toolkits that use a method known as envelope encryption to encrypt your data.

Under this method, KMS generates data keys which are used to encrypt data and are themselves encrypted using your master keys in KMS.

Data keys are not retained or managed by KMS.

AWS services encrypt your data and store an encrypted copy of the data key along with the data it protects.

When a service needs to decrypt your data they request KMS to decrypt the data key using your master key.

If the user requesting data from the AWS service is authorized to decrypt under your master key policy, the service will receive the decrypted data key from KMS with which it can decrypt the your data and return it in plaintext.

All requests to use your master keys are logged in AWS CloudTrail so you can understand who used which key under which context and when they used it.

You can control who manages and accesses keys via IAM users and roles.

You can audit the use of keys via CloudTrail.

KMS differs from Secrets Manager as its purpose-built for encryption key management.

KMS is validated by many compliance schemes (e.g. PCI DSS Level 1, FIPS 140-2 Level 2).

You can perform the following key management functions in AWS KMS:

* Create keys with a unique alias and description.
* Import your own key material.
* Define which IAM users and roles can manage keys.
* Define which IAM users and roles can use keys to encrypt and decrypt data.
* Choose to have AWS KMS automatically rotate your keys on an annual basis.
* Temporarily disable keys so they cannot be used by anyone.
* Re-enable disabled keys.
* Delete keys that you no longer use.
* Audit use of keys by inspecting logs in AWS CloudTrail.
* Create custom key stores\*.
* Connect and disconnect custom key stores\*.
* Delete custom key stores\*.

\* The use of custom key stores requires CloudHSM resources to be available in your account.

Typically, data is encrypted in one of the following three scenarios:

1. You can use KMS APIs directly to encrypt and decrypt data using your master keys stored in KMS.
2. You can choose to have AWS services encrypt your data using your master keys stored in KMS. In this case data is encrypted using data keys that are protected by your master keys in KMS.
3. You can use the AWS Encryption SDK that is integrated with AWS KMS to perform encryption within your own applications, whether they operate in AWS or not.

Custom Key Store:

* The AWS KMS custom key store feature combines the controls provided by AWS CloudHSM with the integration and ease of use of AWS KMS.
* You can configure your own CloudHSM cluster and authorize KMS to use it as a dedicated key store for your keys rather than the default KMS key store.
* When you create keys in KMS you can chose to generate the key material in your CloudHSM cluster. Master keys that are generated in your custom key store never leave the HSMs in the CloudHSM cluster in plaintext and all KMS operations that use those keys are only performed in your HSMs.
* In all other respects master keys stored in your custom key store are consistent with other KMS CMKs.

Key deletion:

* You can schedule a customer master key and associated metadata that you created in AWS KMS for deletion, with a configurable waiting period from 7 to 30 days.
* This waiting period allows you to verify the impact of deleting a key on your applications and users that depend on it.
* The default waiting period is 30 days.
* You can cancel key deletion during the waiting period.

Limits:

* You can create up to 1000 customer master keys per account per region.
* As both enabled and disabled customer master keys count towards the limit, AWS recommend deleting disabled keys that you no longer use.
* AWS managed master keys created on your behalf for use within supported AWS services do not count against this limit.
* There is no limit to the number of data keys that can be derived using a master key and used in your application or by AWS services to encrypt data on your behalf.

# **AWS CloudHSM**

The AWS CloudHSM service helps you meet corporate, contractual and regulatory compliance requirements for data security by using dedicated Hardware Security Module (HSM) instances within the AWS cloud.

AWS and AWS Marketplace partners offer a variety of solutions for protecting sensitive data within the AWS platform, but for some applications and data subject to contractual or regulatory mandates for managing cryptographic keys, additional protection may be necessary.

CloudHSM complements existing data protection solutions and allows you to protect your encryption keys within HSMs that are designed and validated to government standards for secure key management.

CloudHSM allows you to securely generate, store and manage cryptographic keys used for data encryption in a way that keys are accessible only by you.

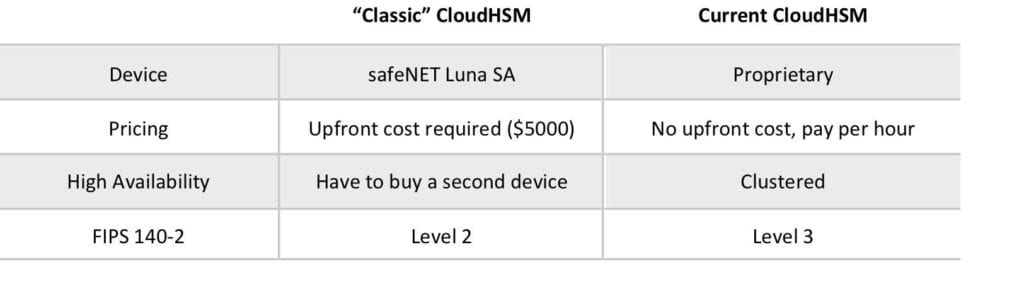
A Hardware Security Module (HSM) provides secure key storage and cryptographic operations within a tamper-resistant hardware device.

HSMs are designed to securely store cryptographic key material and use the key material without exposing it outside the cryptographic boundary of the hardware.

You can use the CloudHSM service to support a variety of use cases and applications, such as database encryption, Digital Rights Management (DRM), Public Key Infrastructure (PKI), authentication and authorization, document signing, and transaction processing.

Runs on a dedicated hardware device, single tenanted.

The table below describes the latest version of CloudHSM and how it differs from its predecessor:



When you use the AWS CloudHSM service you create a CloudHSM Cluster.

Clusters can contain multiple HSM instances, spread across multiple Availability Zones in a region. HSM instances in a cluster are automatically synchronized and load-balanced.

You receive dedicated, single-tenant access to each HSM instance in your cluster. Each HSM instance appears as a network resource in your Amazon Virtual Private Cloud (VPC).

Adding and removing HSMs from your Cluster is a single call to the AWS CloudHSM API (or on the command line using the AWS CLI).

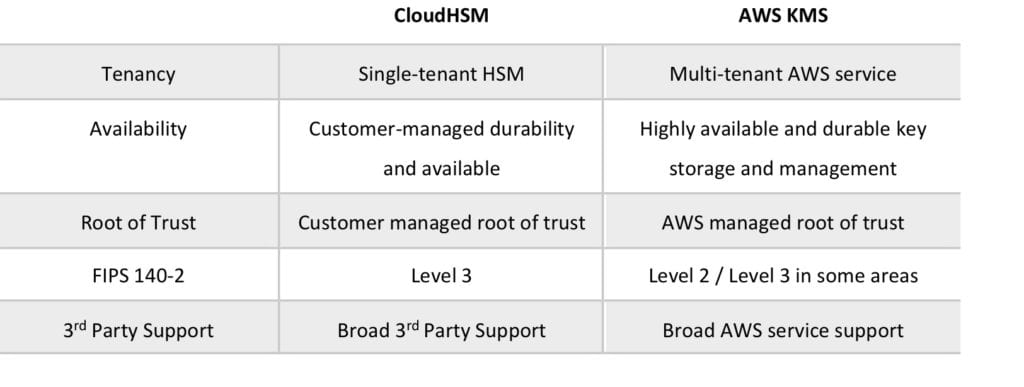
After creating and initializing a CloudHSM Cluster, you can configure a client on your EC2 instance that allows your applications to use the cluster over a secure, authenticated network connection.

Must be within a VPC and can be accessed via VPC Peering.

Applications don’t need to be in the same VPC but the server or instance on which your application and the HSM client are running must have network (IP) reachability to all HSMs in the cluster.

Does not natively integrate with many AWS services like KMS, but instead requires custom application scripting.

Offload SSL from web server, act as an issuing CA, enable TDE for Oracle databases.  
The table below compares CloudHSM against KMS:



# **Amazon Cognito**

Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily.

Amazon Cognito provides authentication, authorization, and user management for your web and mobile apps.

Your users can sign in directly with a user name and password, or through a third party such as Facebook, Amazon, or Google.

The two main components of AWS Cognito are user pools and identity pools:

* User pools are user directories that provide sign-up and sign-in options for your app users.
* Identity pools enable you to grant your users access to other AWS services.

You can use identity pools and user pools separately or together.

AWS Cognito works with external identity providers that support SAML or OpenID Connect, social identity providers (such as Facebook, Twitter, Amazon).

Cognito Identity provides temporary security credentials to access your app’s backend resources in AWS or any service behind Amazon API Gateway.

You can use Amazon, Facebook, Twitter, Digits, Google and any other OpenID Connect compatible identity provider.

You can also integrate your own identity provider.

Cognito exposes server-side APIs.

Users can sign-up and sign-in using email, phone number, or user name.

End users of an application can also sign in with SMS-based MFA.

There is an import tool for migrating users into an Amazon Cognito User Pool.

# **User Pools**

A user pool is a user directory in Amazon Cognito.

With a user pool, users can sign in to your web or mobile app through Amazon Cognito.

Users can also sign in through social identity providers like Facebook or Amazon, and through SAML identity providers.

Whether users sign in directly or through a third party, all members of the user pool have a directory profile that you can access through an SDK.

User pools provide:

* Sign-up and sign-in services.
* A built-in, customizable web UI to sign in users.
* Social sign-in with Facebook, Google, and Login with Amazon, as well as sign-in with SAML identity providers from your user pool.
* User directory management and user profiles.
* Security features such as multi-factor authentication (MFA), checks for compromised credentials, account takeover protection, and phone and email verification.
* Customized workflows and user migration through AWS Lambda triggers.

After successfully authenticating a user, Amazon Cognito issues JSON web tokens (JWT) that you can use to secure and authorize access to your own APIs, or exchange for AWS credentials.

## Identity Pools

Amazon Cognito identity pools (federated identities) enable you to create unique identities for your users and federate them with identity providers.

With an identity pool, you can obtain temporary, limited-privilege AWS credentials to access other AWS services. Amazon Cognito identity pools support the following identity providers:

* Public providers: [**Login with Amazon (Identity Pools)**](https://docs.aws.amazon.com/cognito/latest/developerguide/amazon.html), [**Facebook (Identity Pools)**](https://docs.aws.amazon.com/cognito/latest/developerguide/facebook.html), [**Google (Identity Pools)**](https://docs.aws.amazon.com/cognito/latest/developerguide/google.html).
* Amazon Cognito User Pools.
* [**Open ID Connect Providers (Identity Pools)**](https://docs.aws.amazon.com/cognito/latest/developerguide/open-id.html).
* [**SAML Identity Providers (Identity Pools)**](https://docs.aws.amazon.com/cognito/latest/developerguide/saml-identity-provider.html).
* [**Developer Authenticated Identities (Identity Pools)**](https://docs.aws.amazon.com/cognito/latest/developerguide/developer-authenticated-identities.html).

# **Amazon Cognito Sync**

Amazon Cognito Sync is an AWS service and client library that enables cross-device syncing of application-related user data.

You can use it to synchronize user profile data across mobile devices and the web without requiring your own backend.

The client libraries cache data locally so your app can read and write data regardless of device connectivity status.

When the device is online, you can synchronize data, and if you set up push sync, notify other devices immediately that an update is available.

# **AWS WAF and Shield**

AWS WAF and AWS Shield help protect your AWS resources from web exploits and DDoS attacks.

AWS WAF is a web application firewall service that helps protect your web apps from common exploits that could affect app availability, compromise security, or consume excessive resources.

AWS Shield provides expanded DDoS attack protection for your AWS resources. Get 24/7 support from our DDoS response team and detailed visibility into DDoS events.

We’ll now go into more detail on each service.

## AWS Web Application Firewall (WAF)

AWS WAF is a web application firewall that helps protect your web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources.

AWS WAF helps protect web applications from attacks by allowing you to configure rules that allow, block, or monitor (count) web requests based on conditions that you define.

These conditions include IP addresses, HTTP headers, HTTP body, URI strings, SQL injection and cross-site scripting.

AWS WAF gives you control over which traffic to allow or block to your web applications by defining customizable web security rules.

New rules can be deployed within minutes, letting you respond quickly to changing traffic patterns.

When AWS services receive requests for web sites, the requests are forwarded to AWS WAF for inspection against defined rules.

Once a request meets a condition defined in the rules, AWS WAF instructs the underlying service to either block or allow the request based on the action you define.

With AWS WAF you pay only for what you use.

AWS WAF pricing is based on how many rules you deploy and how many web requests your web application receives.

There are no upfront commitments.

AWS WAF is tightly integrated with Amazon CloudFront and the Application Load Balancer (ALB), services.

When you use AWS WAF on Amazon CloudFront, rules run in all AWS Edge Locations, located around the world close to 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 an Application Load Balancer, your rules run in region and can be used to protect internet-facing as well as internal load balancers.

### **Web Traffic Filtering**

AWS WAF lets you create rules to filter web traffic based on conditions that include IP addresses, HTTP headers and body, or custom URIs.

This gives you an additional layer of protection from web attacks that attempt to exploit vulnerabilities in custom or third party web applications.

In addition, AWS WAF makes it easy to create rules that block common web exploits like SQL injection and cross site scripting.

AWS WAF allows you to create a centralized set of rules that you can deploy across multiple websites.

This means that in an environment with many websites and web applications you can create a single set of rules that you can reuse across applications rather than recreating that rule on every application you want to protect.

### **Full feature API**

AWS WAF can be completely administered via APIs.

This provides organizations with the ability to create and maintain rules automatically and incorporate them into the development and design process.

For example, a developer who has detailed knowledge of the web application could create a security rule as part of the deployment process.

This capability to incorporate security into your development process avoids the need for complex handoffs between application and security teams to make sure rules are kept up to date.

AWS WAF can also be deployed and provisioned automatically with AWS CloudFormation sample templates that allow you to describe all security rules you would like to deploy for your web applications delivered by Amazon CloudFront.

AWS WAF is integrated with Amazon CloudFront, which supports custom origins outside of AWS – this means you can protect web sites not hosted in AWS.

Support for IPv6 allows the AWS WAF to inspect HTTP/S requests coming from both IPv6 and IPv4 addresses.

### **Real-time visibility**

AWS WAF provides real-time metrics and captures raw requests that include details about IP addresses, geo locations, URIs, User-Agent and Referers.

AWS WAF is fully integrated with Amazon CloudWatch, making it easy to setup custom alarms when thresholds are exceeded or particular attacks occur.

This information provides valuable intelligence that can be used to create new rules to better protect applications.

## AWS Shield

AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS.

AWS Shield provides always-on detection and automatic inline mitigations that minimize application downtime and latency, so there is no need to engage AWS Support to benefit from DDoS protection.

There are two tiers of AWS Shield – Standard and Advanced.

## AWS Shield Standard

All AWS customers benefit from the automatic protections of AWS Shield Standard, at no additional charge.

AWS Shield Standard defends against most common, frequently occurring network and transport layer DDoS attacks that target web sites or applications.

When using AWS Shield Standard with Amazon CloudFront and Amazon Route 53, you receive comprehensive availability protection against all known infrastructure (Layer 3 and 4) attacks.

## AWS Shield Advanced

Provides higher levels of protection against attacks targeting applications running on Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator and Amazon Route 53 resources.

In addition to the network and transport layer protections that come with Standard, AWS Shield Advanced provides additional detection and mitigation against large and sophisticated DDoS attacks, near real-time visibility into attacks, and integration with AWS WAF, a web application firewall.

AWS Shield Advanced also gives you 24×7 access to the AWS DDoS Response Team (DRT) and protection against DDoS related spikes in your Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator and Amazon Route 53 charges.

AWS Shield Advanced is available globally on all Amazon CloudFront, AWS Global Accelerator, and Amazon Route 53 edge locations.

Origin servers can be Amazon S3, Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB), or a custom server outside of AWS.

AWS Shield Advanced includes DDoS cost protection, a safeguard from scaling charges as a result of a DDoS attack that causes usage spikes on protected Amazon EC2, Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator, or Amazon Route 53.

If any of the AWS Shield Advanced protected resources scale up in response to a DDoS attack, you can request credits via the regular AWS Support channel.